

UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA INFORMÁTICA



PROYECTO FIN DE CARRERA

**HERRAMIENTA DE CREACIÓN Y EDICIÓN DE
TOPIC MAPS**

TOPIC MAP CREATOR (TMCREATOR)

Autor: Francisco Javier Pérez Diezma

Tutores: Anabel Fraga

Jorge Morato Lara

Página intencionadamente en blanco

Índice

1. INTRODUCCIÓN.....	1
1.1. MOTIVACIÓN DEL PROYECTO.....	2
2. ESTADO DEL ARTE	3
2.1. TOPIC MAP.....	3
2.2. ESTÁNDAR TOPIC MAPS.....	4
2.2.1. Componentes del estándar Topic Maps.....	4
2.3. ESTUDIO DE MERCADO.....	11
2.3.1. TMTab.....	11
2.3.2. Topincs.....	14
2.3.3. Wandora.....	16
2.3.4. TM4L – Editor.....	18
2.3.5. TM4L –Viewer.....	20
2.3.6. TopicMapDesigner.....	22
2.3.7. Ontopia Omnigator.....	24
2.3.8. Conclusiones.....	28
3. HERRAMIENTAS.....	29
3.1. SOFTWARE.....	29
3.1.1. Software de desarrollo.....	29
3.1.2. Software de implantación.....	29
3.2. HARDWARE.....	30
3.2.1. Hardware de desarrollo.....	30
4. ANÁLISIS DEL SISTEMA.....	32
4.1. REQUISITOS DE USUARIO.....	32
4.1.1. Requisitos de capacidad.....	33
4.1.2. Requisitos de restricción.....	38
4.2. MODELO DE CASOS DE USO.....	39
4.3. REQUISITOS SOFTWARE.....	81
4.3.1. Requisitos funcionales.....	82
4.3.2. Requisitos de rendimiento.....	94
4.3.3. Requisitos de interfaz.....	94
4.3.4. Requisitos operacionales.....	96
4.3.5. Requisitos de recursos.....	96
4.3.6. Requisitos de verificación.....	96
4.3.7. Requisitos de pruebas y aceptación.....	96
4.3.8. Requisitos de documentación.....	96
4.3.9. Requisitos de seguridad.....	97
4.3.10. Requisitos de portabilidad.....	97
4.3.11. Requisitos de calidad.....	98
4.3.12. Requisitos de fiabilidad.....	98
4.3.13. Requisitos de mantenimiento.....	98
4.3.14. Requisitos de seguridad (safety).....	98
4.4. DIAGRAMA DE CLASES INICIAL.....	100
4.4.1. Diagrama de paquetes.....	100
4.4.2. Diagrama de clases del paquete controlador.....	101
4.4.3. Diagrama de clases del paquete modelo.....	102
4.4.4. Diagrama de clases del paquete persistencia.....	103
5. DISEÑO DEL SISTEMA.....	104
5.1. DISEÑO ARQUITECTÓNICO.....	104

5.1.1. Descripción de la descomposición.....	105
5.1.2. Descripción de componentes	108
5.2. DISEÑO DETALLADO	122
5.2.1. Modelo.gestionTopicMap	124
5.2.2. Modelo.gestionTopic	138
5.2.3. Modelo.gestionAssociation.....	162
5.2.4. Modelo.gestionScope.....	178
5.2.5. Persistencia	185
6. PLANIFICACIÓN	189
7. ESTIMACIÓN DE COSTES.....	193
7.1. GASTOS DE PERSONAL	194
7.2. GASTOS DE EQUIPOS	195
7.3. GASTOS DE MATERIAL FUNGIBLE.....	196
7.4. GASTOS DE VIAJES Y DIETAS	197
7.5. GASTOS DE LICENCIAS	198
7.6. OTROS GASTOS.....	199
7.7. RESUMEN DEL PRESUPUESTO DEL PROYECTO	200
8. FUTURO	201
9. CONCLUSIONES	202
10. BIBLIOGRAFÍA	206

Índice de Tablas

TABLA 1. URIs DE LOS ELEMENTOS DE TOPINCS.....	14
TABLA 2. MODELO DE ESPECIFICACIÓN DE UN REQUISITO DE USUARIO.....	32
TABLA 3. MODELO DE ESPECIFICACIÓN DE UN REQUISITO SOFTWARE.....	81
TABLA 4. RELACIÓN DE COSTES POR PERSONA Y ACTIVIDAD	194
TABLA 5. GASTOS DE EQUIPOS	195
TABLA 6. GASTOS DE MATERIAL FUNGIBLE.....	196
TABLA 7. GASTOS DE VIAJES Y DIETAS.....	197
TABLA 8. GASTOS DE LICENCIAS	198
TABLA 9. OTROS GASTOS	199
TABLA 10. RESUMEN DEL PRESUPUESTO DEL PROYECTO	200

Índice de Figuras

FIGURA 1. DIAGRAMA DE PAQUETES INICIAL	100
FIGURA 2. DIAGRAMA DE CLASES INICIAL DEL PAQUETE CONTROLADOR.....	101
FIGURA 3. DIAGRAMA DE CLASES INICIAL DEL PAQUETE MODELO	102
FIGURA 4. DIAGRAMA DE CLASES INICIAL DEL PAQUETE PERSISTENCIA.....	103
FIGURA 5. MVC DE SUN MICROSYSTEMS	104
FIGURA 6. ARQUITECTURA MVC.....	105
FIGURA 7. ARQUITECTURA DEL SISTEMA	106
FIGURA 8. ARQUITECTURA DEL MODELO	107
FIGURA 9. COMPONENTE GESTIONTOPICMAP	109
FIGURA 10. TAREAS Y TIEMPOS PLANIFICADOS INICIALMENTE	190
FIGURA 11. DIAGRAMA DE GANTT INICIAL	191
FIGURA 12. TAREAS Y TIEMPOS REALES.....	192



1. Introducción

Actualmente la Web está evolucionando hacia un tipo de Web extendida, que está dotada de un mayor significado, ayudando al usuario a encontrar respuesta a cada una de sus preguntas de una forma rápida y sencilla. A esta evolución se le ha denominado Web Semántica.

El principal objetivo de la Web Semántica es hacer de la Web un medio más colaborativo y más entendible. Para conseguir dicho objetivo la Web Semántica se apoya en nuevas técnicas de inteligencia artificial y nuevas tecnologías web, desarrollando herramientas y sistemas que permitan compartir, descubrir y acceder al conocimiento.

A partir del enfoque descrito anteriormente surgen gran variedad de herramientas, sistemas y lenguajes de programación. Estas tecnologías son utilizadas para asignar semántica a documentos, estructuras y objetos de la web, así como aportar a las computadoras la posibilidad de entender e interpretar cada uno de ellos.

De todas las tecnologías que permiten dar semántica a documentos, estructuras, etc., nos centraremos en los Topic Maps, cuyo inicio es planteado como una respuesta a la necesidad de establecer estructuras de índices.

Las principales ventajas que ofrecen los Topic Maps para la representación del conocimiento en la web semántica se pueden resumir de la siguiente forma:

- Los Topic Maps pueden dar una semántica a elementos que están en la web, al organizarlos y describirlos.
- Presencia de perfiles de usuarios que permiten adaptarse a distintas comunidades para compartir recursos de información.
- Navegabilidad e inferencia mediante estructuras semánticas: Mejora no sólo la recuperación de información, sino también la gestión del conocimiento y el mantenimiento de los Topic Maps.
- Fusión con otras estructuras de conocimiento, que permite una gestión descentralizada.
- Buena escalabilidad y compatibilidad.

Al igual que Topic Map presenta una serie de ventajas, también tiene una serie de desventajas que en la actualidad se están intentando mejorar. Algunos de dichos aspectos a mejorar son: el desarrollo de lenguajes para realizar consultas a los Topic Maps, herramientas de visualización gráfica de estructuras de tópicos o herramientas de generación de Topic Map automáticos.

Este proyecto se va a centrar en la creación de una herramienta que permita generar Topic Maps de una forma automática, disminuyendo de esta forma el trabajo manual que hoy se requiere en la escritura de código de marcado de estructuras como es el XTM, a la herramienta a la que se denominará TMCreator.



1.1. Motivación del proyecto

La motivación del trabajo a desarrollar tiene su punto de partida en la necesidad del mercado actual de una herramienta que permita crear ficheros XTM de una forma sencilla y que permita tener el control de todos los componentes del Topic Map.

El objetivo de este proyecto es la elaboración de una herramienta en entorno java que sirva como apoyo para aquellas personas que deseen aprender cómo se crea un Topic Map; por lo tanto, si el proyecto debe servir para que los usuarios aprendan el estándar de Topic Map, la herramienta debe permitir al usuario tener el control de la creación de todos los elementos que formen parte del Topic Map.

La mayoría de las herramientas que permiten crear Topic Map, tienen parte del trabajo automatizado, ocultando información relevante al usuario que desea conocer exactamente cómo se debe crear cada elemento de Topic Map; por esta razón, se ha estudiado la posibilidad de crear la herramienta TMCreator que ayude a los usuarios a realizar Topic Map de una forma más sencilla mediante una interfaz gráfica, pero que no realice ninguna operación automática que oculte información al usuario, obligando a éste a generar el fichero xtm y estudiar cómo la herramienta lo ha construido.

Otro aspecto negativo de las herramientas actuales es la complejidad de los xtm que generan. Los Topic Map tienen gran cantidad de relaciones realizadas mediante un identificador del elemento; este identificador es generado automáticamente por el sistema dificultando gravemente el entendimiento y la navegabilidad del xml, ya que suelen ser cadenas extensas de números y letras poco manejables. Por esta razón, se considera necesaria la creación de una herramienta que permita al usuario insertar un identificador del elemento que resulte más entendible; de esta forma el xtm generado será más simple y entendible sin necesidad de estar realizando búsquedas de identificadores poco manejables.

2. Estado del arte

Este apartado está compuesto principalmente por dos partes. En la primera parte intentaremos explicar el estándar de Topic Map, mientras que en la segunda parte se realizará un estudio del mercado actual.

2.1. Topic Map

El estándar de Topic Map nace en 1991 fruto del trabajo realizado por el llamado Grupo de Davenport, grupo que surge como un consorcio de empresas. El grupo se dividió en dos subgrupos en 1993: uno de ellos se centró en la definición de una DTD para el contenido de manuales, dando como resultado la DTD DocBook; el otro subgrupo (Conventions for the Application of HyTime) tenía como objetivo principal desarrollar un índice común a partir de las distintas fuentes de documentación sobre X-Windows que sirviera como base para otros nuevos índices.

El trabajo de este grupo resultó ser más complejo de lo esperado, por lo que todos los esfuerzos del grupo se centraron en entender la semántica de los índices de los libros para posteriormente adaptarla en los ordenadores

En 1996 el grupo de trabajo de SGML de la ISO aceptó el borrador desarrollado, denominado entonces “Topic Navigation Maps”, como nuevo documento de trabajo. Finalmente, en el verano de 1999 fue aceptado como norma y publicada como ISO/IEC 13250:2000 Topic Maps (International Organization for Standardization, 2000) [1].

El estándar está descrito en lenguaje HyTime (Hypermedia/Time-based Structuring Language), el cual, a su vez, es una aplicación del lenguaje SGML (Standard Generalized Markup Language) para procesar documentos. Esta sintaxis de intercambio es muy flexible, lo que permite a los usuarios escribir sus propias DTDs y el uso de una gran variedad de estructuras de enlace multimedia definidas mediante HyTime. Sin embargo, el número de personas que dominan este lenguaje no es elevado, pues a pesar de su flexibilidad, o a causa de ello, resulta complejo.

A comienzos de 2000 se funda una organización independiente, TopicMaps.Org, con el objetivo de adaptar esta norma a un lenguaje más adecuado para la Web. Basándose en las recomendaciones XML y el estándar para enlaces XLink del W3C, publicó la primera versión de una DTD para expresar Topic Maps en marzo de 2001, conocida como XTM.

La especificación XTM 1.0. (TopicMaps.Org, 2001) fue admitida por la ISO e incorporada al estándar mediante una Enmienda Técnica en octubre de ese mismo año. En mayo de 2002 el Subcomité 34 del JTC1 de la ISO/IEC (Join ISO/ICE Technical Comité SC34) aprobó y publicó la segunda edición de la norma, recogida finalmente por la ISO como ISO/IEC 13250:2003. Topic Maps (International Organization for Standardization, 2003), incorporando en su Anexo C la sintaxis XML [2].

El modelo intenta proporcionar un esquema de representación de estructuras de conocimiento, en forma de red semántica, y asociarlas con recursos de información, y que incorpore un valor añadido similar al que un índice analítico añade a un libro.

La norma especifica no sólo los conceptos que conforman los Topic Maps sino también el formato de intercambio estándar de Topic Maps entre diferentes aplicaciones.

Los Topic Maps enlazan conceptos mediante asociaciones, en principio sin limitación alguna, lo que los asemeja a otros instrumentos (como tesauros, mapas conceptuales u ontologías, por ejemplo) y permite que sean utilizados para modelar y expresar cualquiera de ellos.

A continuación desarrollaremos los conceptos que componen este modelo. Para ello, nos centraremos en la especificación XTM 1.0.

2.2. Estándar Topic Maps

2.2.1. Componentes del estándar Topic Maps

El estándar ISO/IEC 13250:2003 está constituido por tres elementos básicos: Topics, Association, y Occurrence.

Topics

El elemento principal de los Topic Maps es el *Topic*, el cual constituye la representación material de un *subject*, entendiendo por subject cualquier tema sobre el que pueda hablarse o ser concebido por un ser humano.

Así, el término topic indica una parte del Topic Map que representa al subject al que está representando. Por lo general, en un mapa de tópicos cada subject es representado por un único tópico y viceversa.

Un topic puede representar cualquier tema: personas, entidades individuales o colectivas, conceptos, etc.

Cada topic se puede asociar con otros topics denominados *clases topic types*, por ejemplo un topic “dedo” puede tener un topic type que sea “mano”. Los topic types definen relaciones clase-instancia.

A continuación se expone un ejemplo de Topic Type:

```
<topic id="tempestad">
  <instanceOf>
    <topicRef xlink:href="#obra">
  </instanceOf>
</topic>
```

En el ejemplo se hace referencia a la obra *Tempestad* de William Shakespeare y se observa como el topic con *id="tempestad"* está relacionado con el topic obra.

Como los Topics representan un subject, es posible asignar a cada topic unas características propias. Cada topic dispone de las siguientes características:

- Nombre de topic (topic name).
- Apariciones (topic occurrence).
- Rol como miembro de una asociación (role associations).

Esta asignación de características se considera válida para un determinado contexto. Dos topics con las mismas características se consideran idénticos, produciendo duplicidad de datos.

A través del *topic name* se permite al usuario dar un nombre legible y entendible para el ser humano. Ya que un mismo concepto puede ser designado con gran variedad de nombres, el estándar permite disponer de varios nombres mediante la asignación de múltiples *base names*. Adicionalmente, el base name puede incluir sinónimos (*variant name*).

Siguiendo con el ejemplo expuesto anteriormente de las obras de William Shakespeare añadimos un *topic name*:

```
<topic id="tempestad">
  <instanceOf>
    <topicRef xlink:href="#obra">
  </instanceOf>
  <baseName>
    <baseNameString>La Tempestad</baseNameString>
  </baseName>
</topic>
```

Occurrence

Son elementos externos de información, enlazados al topic mediante una referencia que sirve para su localización y que aportan información al topic.

Las referencias a recursos se realizan, en XTM, a través de URIs (Uniform Resource Identifiers). Dichos recursos no son obligatorios en la creación de un topic, por lo que un topic puede tener desde cero a n ocurrences.

Al igual que los topics, cada *occurrence* puede relacionarse con una instancia de una clase de occurrence (denominada también *occurrence type*), que puede o no indicarse de forma explícita. En XTM se expresa mediante el elemento *instanceOf*. El occurrence type es un topic.

Para ejemplificar la creación de un occurrence dentro de un topic y continuando con el ejemplo anterior de las obras de William Shakespeare, se expone a continuación el siguiente cuadro:

```
<topic id="tempestad">
  <instanceOf>
    <topicRef xlink:href="#obra">
  </instanceOf>
  <baseName>
    <baseNameString>La Tempestad</baseNameString>
  </baseName>
  <occurrence>
    <instanceOf>
      <topicRef xlink:href="#formato-texto-plano">
    </instanceOf>
    <resourceRef xlink:href="ftp://www.gutenberg.org/pub/gutenberg/etext97/1ws4110.txt">
  </occurrence>
</topic>
```

Association

En el estándar la *association* es definida como “una relación entre uno o más topics, donde cada uno de ellos juega un rol como miembro de dicha asociación”.

La *association* en el modelo Topic Maps no indica el sentido en el que se tiene que interpretar la asociación; para poder expresar el sentido de la asociación es necesario introducir el concepto de *association role*.

Del mismo modo que ocurría con los topics y las occurrences, las *associations* se pueden clasificar en *association types*. La *association type* es el verbo que une a los topics y también, al igual que las occurrences, es un topic.

En la especificación de Topic Map vienen incluidas dos clases de *association*. Estas *associations* tienen definidos los subjects correspondientes mediante *published subjects*: la *class-instance association* (asociación clase-instancia) y la *superclass-subclass association* (asociación superclase-subclase).

La *superclass-subclass association* permite construir jerarquías de clases, aplicadas tradicionalmente para realizar taxonomías o clasificaciones de materias.

La principal diferencia entre estos dos tipos de *associations* es que la relación superclase-subclase es transitiva, es decir, que las propiedades de la clase superior se van heredando por las subclases inferiores a ella; mientras que en la relación clase-instancia no es transitiva, es decir, las propiedades de la clase no son heredadas por las instancias.

A continuación se expone un ejemplo de creación de una Association:

```
<association>
  <instanceOf><topicRef xlink:href="#es-autor-de"/></instanceOf>
  <member>
    <roleSpec><topicRef xlink:href="#obra"/></roleSpec>
    <topicRef xlink:href="#tempestad"/>
  </member>
  <member>
    <roleSpec><topicRef xlink:href="#autor"/></roleSpec>
    <topicRef xlink:href="#william-shakespeare"/>
  </member>
</association>
```

Otros componentes de Topic Maps

El modelo Topic Maps incorpora, además de los tres pilares descritos anteriormente, otros elementos que le dan al modelo una potencia mayor. En un principio, los elementos que permiten a la especificación tener un mayor potencial eran *scope*, *facet* y *public subject*, pero cuando se instauró la especificación de XTM el elemento *facet* fue eliminado.

El elemento *facet* proporcionaba mecanismos para asignar recursos o fuentes de información. Los facets eran usados normalmente para suplir la clase de metadatos que puede ser provista por atributos SGML o XML.



El elemento *scope* lo constituyen los límites de validez de las características asignadas a un topic. El concepto se añadió al modelo Topic Map para atender al hecho de que no hay una única visión del mundo, debido a razones diversas, como por ejemplo los distintos idiomas o las culturas.

Por último, el elemento *public subject* proporciona la posibilidad de fusionar Topic Maps entre sí.

DTD - XTM 1.0

Para completar la descripción de este estándar, a continuación se explicará la estructura de la DTD para la sintaxis XML, XTM 1.0 Topic Maps. La versión completa original puede consultarse y enlazarse como DTD [9].

El elemento raíz de esta DTD es el elemento o etiqueta topicMap. Éste puede incluir sólo uno de los subelementos (o subetiquetas) topic, association, mergeMap, como indica la barra vertical, aunque pueden repetirse en cualquier orden cuantas veces se desee (indicado por el asterisco tras el paréntesis).

```
<!ELEMENT topicMap ( topic | association | mergeMap )*>
<!ATTLIST topicMap
  id          ID          #IMPLIED
  xmlns       CDATA       #FIXED 'http://(esta DTD en español no posee URI)'
  xmlns:xlink CDATA       #FIXED 'http://www.w3.org/1999/xlink'
  xml:base    CDATA       #IMPLIED
>
```

Un ejemplo del comienzo de un documento XML que exprese un Topic Map es el siguiente:

```
<topicMap
  id="Ejemplo"
  xmlns = 'http://www.topicmaps.org/xtm/1.0/'
  xmlns:xlink = 'http://www.w3.org/1999/xlink'
  ...
</topicMap>
```

El elemento topic debe estar identificado internamente mediante un identificador. Dicho identificador es obligatorio y único, para poder servir como referencia de enlace desde cualquier otro lugar. Incluye los subelementos instanceOf, subjectIdentity, baseName y occurrence:

```
<!-- topic: elemento Topic..... -->
<!ELEMENT topic ( instanceOf*, subjectIdentity?, ( baseName | occurrence )* )>
<!ATTLIST topic
  id          ID          #REQUIRED
>
<!-- instanceOf: Apunta a un Topic representando una clase ... -->
<!ELEMENT instanceOf (topicRef | subjectIndicatorRef)>
<!ATTLIST instanceOf
  id ID #IMPLIED
>
<!-- subjectIdentity: Subject representado por el Topic .... -->
<!ELEMENT subjectIdentity
(resourceRef?, (topicRef | subjectIndicatorRef)* )>
```



```
<!-- subjectIdentity
id ID #IMPLIED
-->
<!-- topicRef: Referencia a un elemento Topic..... -->
<!-- ELEMENT topicRef EMPTY -->
<!-- ATTLIST topicRef
id ID #IMPLIED
xlink:type NMTOKEN #FIXED 'simple'
xlink:href CDATA #REQUIRED
-->
<!-- subjectIndicatorRef: Referencia a un indicador de Subject-->
<!-- ELEMENT subjectIndicatorRef EMPTY -->
<!-- ATTLIST subjectIndicatorRef
id ID #IMPLIED
xlink:type NMTOKEN #FIXED 'simple'
xlink:href CDATA #REQUIRED
-->
```

Todos los subelementos que componen los elementos `instanceOf` y `subjectIdentity` constituyen, en definitiva, elementos de enlace o referencias internas o externas. Se trata de elementos que se van repitiendo a su vez en otros, realizando la misma función.

Mientras el ID de un `topic` sirve como representación interna de éste, el nombre de base, que puede indicarse o no dentro del elemento `baseName`, constituye una representación externa.

```
<!-- baseName: Nombre Base o preferente de un Topic .....-->
<!-- ELEMENT baseName ( scope?, baseNameString, variant* ) -->
<!-- ATTLIST baseName
id ID #IMPLIED
-->
<!-- ELEMENT baseNameString ( #PCDATA ) -->
<!-- ATTLIST baseNameString
id ID #IMPLIED
-->
<!-- ELEMENT variant ( parameters, variantName?, variant* ) -->
<!-- ATTLIST variant
id ID #IMPLIED
-->
<!-- ELEMENT variantName ( resourceRef | resourceData ) -->
<!-- ATTLIST variantName
id ID #IMPLIED
-->
<!-- ELEMENT parameters ( topicRef | subjectIndicatorRef )+ -->
<!-- ATTLIST parameters
id ID #IMPLIED
-->
<!-- ELEMENT scope ( topicRef | resourceRef | subjectIndicatorRef )+ -->
<!-- ATTLIST scope
id ID #IMPLIED
-->
```

En el elemento *baseNameString* es donde se recoge el nombre base, que es único. El elemento *variant*, repetible y con varios subelementos, indica sus variantes posibles; en este último caso, pueden indicarse incluso variantes de variantes. El elemento *scope* es opcional, pero de un solo uso y compuesto de subelementos.

El subelemento *occurrence* del elemento topic, está definido de la siguiente forma:

```
<!-- occurrence: Recursos considerados como una Occurrence . -->
<!ELEMENT occurrence instanceOf?, scope?, (resourceRef | resourceData)>
<!ATTLIST occurrence
    id ID #IMPLIED
>

<!ELEMENT resourceRef EMPTY >

<!ATTLIST resourceRef
    id ID #IMPLIED
    xlink:type NMTOKEN #FIXED 'simple'
    xlink:href CDATA #REQUIRED
>

<!ELEMENT resourceData ( #PCDATA )>
<!ATTLIST resourceData
    id ID #IMPLIED
>
```

ResourceRef y *resourceData* permiten enlazar los recursos e incluir metadatos de distinto tipo.

El subelemento *association* del elemento topicMap puede incorporar los subelementos *instanceOf*, *scope* y *member*, debiendo aparecer éste último al menos una vez, puesto que define el tipo de participación del topic en la relación.

```
<!-- association: Relación entre Topics ..... -->
<!ELEMENT association (instanceOf?, scope?, member+)>
<!ATTLIST association
    id ID #IMPLIED
>

<!ELEMENT member (roleSpec?, (topicRef|resourceRef|subjectIndicatorRef)*)>
<!ATTLIST member
    id ID #IMPLIED
>

<!-- roleSpec: Apunta a un Topic que funciona describiendo el rol realizado en la relación (Association Role) ..... -->
<!ELEMENT roleSpec (topicRef | subjectIndicatorRef) >
<!ATTLIST roleSpec
    id ID #IMPLIED
>
```

Es de destacar el subelemento *roleSpec* de *member* dado que es el que, enlazando con otro topic que lo indica, describe el papel desempeñado por el topic en la relación.

El subelemento *mergeMap* del elemento topicMap, por último, es el que facilita la unión de mapas haciendo referencia al que quiere incorporarse. Es el mecanismo que permite la



actualización para la incorporación de nuevos recursos o de la estructura de conocimiento del propio mapa. Es posible únicamente a través de una aplicación que procese este elemento.

```
<!ELEMENT mergeMap ( topicRef | resourceRef | subjectIndicatorRef )* >
<!--ATTLIST mergeMap
id ID #IMPLIED
xlink:type NMTOKEN #FIXED 'simple'
xlink:href CDATA #REQUIRED
-->
```

Como puede observarse, el modelo Topic Map está constituido por un pequeño número de conceptos, lo que le da apariencia de simplicidad. Sin embargo, la combinación de éstos permite la descripción de estructuras altamente complejas, haciéndolo apropiado para su uso en numerosas aplicaciones.

2.3. Estudio de mercado

En el presente apartado se realizará un estudio de las herramientas existentes en el mercado actual y que tengan funcionalidades semejantes a la herramienta que se desea crear. Además se estudiarán algunas de las herramientas de visualización de Topic Maps del mercado ya que, aunque no es el objetivo principal del producto que se desea desarrollar, se deberá representar la información de un Topic Map de una forma sencilla para el usuario.

2.3.1. TMTab

2.3.1.1 Descripción

TMTab [3] es un plugin de la herramienta Protege que permite construir ontologías y exportarlas como un Topic Map en formato XTM.

El plugin permite crear TM a partir de un fichero inicial que se carga en el sistema. El fichero es una ontología base preparada para la creación de Topic Maps.

La página web en la que se puede encontrar más información sobre TMTab es la siguiente: <http://www.techquila.com/tmtab/index.html>

2.3.1.2 Funcionalidad

Topic Map

Gracias al fichero que es cargado inicialmente, TMTab permite crear Topic Map básicos de una forma sencilla. Para comenzar, primero se debe crear un nuevo proyecto vacío de Protege e incluir la ontología base que podemos encontrar en el siguiente path “<tu directorio protege>/projects/topicmaps/topicmap.pprj”. Con esta acción se carga la clase Topic y todas sus subclases, quedando el sistema preparado para comenzar a crear nuestro Topic Map.

Topics

La herramienta permite crear nuevos topics. Para ello, hay que seleccionar la instancia de la clase Topic del árbol que aparece a la izquierda de la ventana, y hay que pulsar ‘C’ para crear un nuevo topic. Una vez creado el topic se pueden rellenar las siguientes características:

- *Model Name*: usado para añadir un nombre por defecto del topic. Este nombre será generado como un baseName.
- *Subject*: en este elemento se puede especificar la URI que identifica el subject del topic.
- *Subject Indicators*: semejante al componente *subject*, pero se pueden indicar una lista de URIs.
- *Names*: lista de instancias de la clase Name.
- *Occurrences*: lista de instancias de la clase Occurrences
- *Types*: lista de instancias de la clase Topic.

Names

Los Names son creados como instancias de la clase Name. La clase Name es una clase derivada de Topic, por lo que además de los atributos propios dispone de los atributos explicados en el apartado anterior que correspondían a la clase Topic. Los atributos propios de la clase Name son:

- *Name String*: usado para especificar el nombre.
- *Scope*: lista de instancias de Topic que definen el scope del nombre.
- *Variants*: lista de instancias de la clase Variant-Name.

La clase Variant-Name es usada para crear variantes de las instancias de la clase Name. Al igual que la clase Name, la clase Variant-Name es una clase que deriva de Topic. La clase Variant-Name además de los atributos de la clase Topic dispone de los siguientes atributos:

- *Name Resources*: lista de strings, que deberían ser URIs, que pueden ser usadas como variant name.
- *Name String*: string que puede ser usado como variant name.
- *Scope*: lista de instancias de la clase Topic. Es usada para definir los “parámetros” del variant.

Occurrences

Las Occurrences de los topics son creadas como instancias de la clase Occurrence. Al igual que la clase Name, la clase Occurrence es una clase que deriva de Topic, por lo que además de los atributos de la clase Topic dispone de los siguientes atributos:

- *Resource*: es un string que especifica la información de la Occurrence. Se suele usar para insertar metadatos en el topic.
- *Resource Locator*: string que especifica la dirección de la información de la occurrence. El valor suele ser la dirección de una página web.

Association

Las asociaciones entre topics son creadas a partir de las instancias de la clase Association. El único parámetro de la clase Association es el parámetro Members. Este parámetro acepta una lista de instancias de la clase Member. Así mismo, la clase Member es usada para definir cada rol de los miembros de la asociación y los topics que tienen ese rol. La clase Member dispone de los siguientes parámetros adicionales:

- *Role players*: lista de instancias de la clase Topic. Dicha lista especifica los topics que tiene el rol definido por la instancia Member en la asociación.



2.3.1.3 Dependencias

TMTab ha sido construido como un plugin de Protégé por lo que depende totalmente de él. Además, el plugin hace uso de la librería de procesamiento de topic map TM4J y del parser de XML Xerces.

2.3.1.4 Licencia

TMTab actualmente se está distribuyendo gratuitamente mediante un zip en el que podemos encontrar los archivos binarios del sistema.

2.3.2. Topincs

2.3.2.1 Descripción



Topincs [4] es una herramienta que permite crear Topic Maps a través de Firefox y Opera. La herramienta consta de un cliente que permite realizar los cambios en el Topic Map y un servidor que los almacena.

Topincs dispone de todas las funcionalidades necesarias para crear un Topic Map a través de Internet o en una intranet.

La página web en la que se puede encontrar más información sobre la herramienta es la siguiente: <http://www.cerny-online.com/topincs/>

2.3.2.2 Funcionalidad

Topincs permite crear todos los ítems que aparecen en el modelo de datos de Topic Map. Estos ítems son asignados a una URI pudiendo manipularse gracias a los métodos http: *put*, *get*, *post* y *delete*.

El uso de las funciones http no es demasiado complicado; por ejemplo, si se quiere crear un topic en un Topic Map, se debe hacer “*post*” del topic sobre la URI del Topic Map. Esta función devolverá la URI bajo la que el nuevo topic queda registrado. Otro ejemplo práctico: si se quiere añadir un name a un topic, se recupera el topic mediante la URI devolviendo el nombre. Si se desea cambiar el nombre se usa la función “*put*” con la nueva versión del nombre.

Las URIs cuya función es identificar elementos siguen una forma específica. Por defecto, la instalación de Topincs se almacena en una dirección del siguiente estilo: *http://yourhost/topincs/*. Los Topics Maps se localizarán en *http://myhost/topincs/maps/* y los demás elementos se podrán encontrar en las siguientes direcciones:

Elemento	Dirección
Topic map	<i>http://yourhost/topincs/topicmaps/yourmap/</i>
Topic	<i>http://yourhost/topincs/topics/1</i>
Name	<i>http://yourhost/topincs/names/1</i>
Variants	<i>http://yourhost/topincs/variants/1</i>
Occurrences	<i>http://yourhost/topincs/occurrences/1</i>
Association	<i>http://yourhost/topincs/associations/1</i>
Roles	<i>http://yourhost/topincs/roles/1</i>

Tabla 1. URIs de los elementos de Topincs



La versión actual de Topincs tiene como problema principal que los Variants no son soportados. No obstante, en la página web de la herramienta aparece un compromiso por su parte de que, en versiones posteriores, se subsanará esta limitación.

2.3.2.3 Dependencias

El funcionamiento de la actual implementación de Topincs depende de Apache, MySQL y PHP. Además, necesita una máquina para alojar el servidor, la cual puede disponer de distintos sistemas operativos: Windows, Linux o Mac OS X.

Topincs es una aplicación cliente servidor, lo que implica que depende del protocolo http, y por consiguiente, que el cliente podrá comunicar con el servidor siempre y cuando el protocolo http no sufra modificaciones importantes.

2.3.2.4 Licencia

El uso de Topincs está limitado a usuarios privados y usuarios con propósitos académicos. Una vez adquirida la licencia del sistema, se permite cambiar el código, pero no distribuir los cambios.

2.3.3. Wandora

2.3.3.1 Descripción



El propósito general de Wandora [5] es la extracción, manejo y publicación de conocimiento basado en Topic Maps. Wandora es una aplicación construida en Java, utilizada para construir y manejar Topic Maps.

Wandora es una aplicación similar a Protege o TM4L, pero la forma de uso es mas similar a un navegador Web. El modelo de Topic Map de Wandora es un poco limitado si se compara con el estándar. Wandora soporta almacenar en una base de datos SQL los topics

maps.

La página web en la que se puede encontrar más información sobre la herramienta es la siguiente: http://www.wandora.org/wandora/wiki/index.php?title=Main_Page

2.3.3.2 Funcionalidad

Wandora usa una versión de Topic Map ligeramente reducida. Las partes del estándar que no están recogidas por la aplicación se pueden expresar de una forma distinta.

Las diferencias más importantes son:

1. Subject identifiers y subject locators nunca se resuelven; son simplemente usados como URI a un documento local o un documento online.
2. Sólo se permite un Name por cada topic, además los base names no pueden contener Scope.
3. Los variants no son dependientes de los base names, pero todos son considerados variants del single base name del topic. Los variant pueden llevar Scope.
4. En las occurrence no se pueden insertar referencias, sólo datos. Cuando es necesario insertar una referencia en una occurrence, puede efectuarse creando un nuevo topic para el URI que deseamos insertar. Este URI se utilizará como el subject locator del topic; posteriormente, se deberá realizar una asociación entre ambos topic.
5. Las occurrences pueden tener solamente un scope de un topic, lo cual se usa normalmente para almacenar el lenguaje en el que está escrita la occurrence.
6. Los topic pueden contener una occurrence por tipo y scope.
7. Los topic pueden contener un variant por scope.
8. Las associations no pueden tener scope.
9. Associations pueden tener un usuario por miembro y no pueden tener dos miembros con el mismo rol.



2.3.3.3 Dependencias

No se han encontrado dependencias importantes; simplemente se debe indicar que necesita una máquina virtual de Java para poder funcionar correctamente.

2.3.3.4 Licencia

El software es propiedad de Grip Studios Interactiva Oy. El copyright limita al uso personal, no comercial, con licencia gratuita para descargar, instalar y usar el software Wandora en el ordenador del usuario.

2.3.4. TM4L – Editor

2.3.4.1 Descripción



TM4L [6] Editor es un editor de ontologías que permite al usuario construir ontologías usando Topic Maps. El editor TM4L permite la fusión fácil y eficaz de los recursos existentes de la información, a la vez que mantiene su estructura significativa. Esta característica permite añadir flexibilidad y posibilita la reutilización de los repositorios existentes. La salida del editor sigue el estándar de Topic Maps de XML (XTM), por lo que es compatible con cualquier herramienta estándar de XTM.

Los principales objetos que manipula el editor TM4L son topics, relaciones entre topics, resources y contexts. El editor incluye cuatro secciones diferentes: Topic Map, Topics, Relationships y Themes. En el siguiente apartado se describirán más detalladamente las funcionalidades del editor.

La página web en la que se puede encontrar más información sobre la herramienta es la siguiente: <http://compsci.wssu.edu/iis/nsdl/download.html>

2.3.4.2 Funcionalidad TM4L - Editor

Topic Map

El editor permite crear nuevos Topic Maps seleccionando la pestaña Topic Map en el mismo. En la sección que permite crear un nuevo Topic Map aparecerán una serie de parámetros a rellenar que son opcionales, a excepción del parámetro Subject, el cual es utilizado para dar una referencia del nuevo Topic Map. Otros parámetros que aparecen son: título del TM, creador, descripción, publicación, fecha, etc.

Topics

En la sección de Topics el usuario puede crear, editar y eliminar topics. Cada vez que se crea un nuevo topic se debe rellenar la siguiente información: subject indicator, names, types y related resources. Además, a cada nuevo topic se le asignará automáticamente un Id.

Categorías de Topic: TM4L soporta dos categorías distintas para crear topics: *domain ontology topics* y *utility topics*. La primera categoría de topics es definida por el usuario; la segunda categoría es generada automáticamente por el editor.

TM4L utiliza las siguientes categorías de Topic: *association types*, *association role types*, *occurrence types*, *name use types*, y *themes*.

Topic names: TM4L permite múltiples topic names. Uno de esos names será el principal y los demás serán nombres alternativos. Cada nombre puede tener sus name variant para poder usarse con algún propósito en especial.

En TM4L se permite 4 nombres: sort, search, display y draw.

Topic Types: TM4L permite múltiples topic types. El usuario dispone de dos formas de declarar un topic type: automáticamente seleccionando un topic existente antes de la creación del nuevo topic, o manualmente añadiendo un padre en el *Parent Topic Panel*.

Resources: TM4L soporta dos tipos de resources: los resources internos y los resources externos. Los internos suelen ser piezas de información sobre un concepto, como por ejemplo una definición. Los resources externos pueden ser cualquier objeto Web que es referenciado mediante su URI.

Associations

Cada asociación está formada por un *type* y por uno o más *members*. El sistema dispone de una serie de *types* predefinidas que el usuario puede usar. En la pestaña *Relationships*, el usuario puede definir *types* y roles; crear relaciones especificando sus *types*, roles y miembros; y editar y borrar relaciones.

El usuario, al definir una relación, selecciona todos los atributos implicados (*types*, *members*, y *roles* de la relación) de una serie de listas. Si se desea definir el alcance de la relación, se deberá añadir en la sección *Theme*.

Un nuevo punto a destacar es que el sistema permite adoptar tres perspectivas del concepto básico, permitiendo modelar diferentes clasificaciones para un mismo *Topic*.

Themes

El concepto *Theme* permite al usuario definir un contexto, dentro del cual algunas características del *Topic* son válidas.

Algunos de los componentes que hacen uso de “*Themes*” son:

- Topic Names
- Topic Resources
- Type de las asociaciones

2.3.4.3 Dependencias

No se han encontrado dependencias importantes; simplemente se debe indicar que necesita máquina virtual de Java, ya que el sistema está implementado con dicho lenguaje.

2.3.4.4 Licencia

Es un software de libre distribución.

2.3.5. TM4L –Viewer

2.3.5.1 Descripción



TM4L Viewer [6] es un visor de Topic Map que permite al usuario realizar búsquedas y visualizar el contenido de un Topic Map de una forma sencilla y rápida.

TM4L Viewer soporta múltiples vistas. La interfaz ofrece al usuario diferentes perspectivas de una entidad previamente seleccionada.

La finalidad de TM4L Viewer es proporcionar al usuario un interfaz gráfico intuitivo para la navegación. Para conseguirlo dispone de tres vistas: gráficos, árbol y texto.

La página web en la que se puede encontrar más información sobre la herramienta es la siguiente: <http://compsci.wssu.edu/iis/nsdl/download.html>

2.3.5.2 Modos del visualizar los Topic Map

Modo gráfico:

Este modo permite visualizar toda la información relativa al Topic Map seleccionado mediante objetos gráficos. Una desventaja de este modo es que cuando se dispone de mucha información relacionada, la visualización es bastante compleja; por esta razón, el modo gráfico no muestra los recursos asociados al topic.

Modo texto:

En este modo el sistema muestra las características de un topic o de una relación en forma de texto.

Modo árbol:

El modo árbol permite al usuario visualizar toda la información relacionada con el Topic Map seleccionado. En la mayoría de los casos, el resultado será significativo, pero hay algunos casos en los que el resultado obtenido no es correcto.

2.3.5.3 Funcionalidad TM4L-Viewer relacionado con Topic Map

Abrir y cerrar un Topic Map

TM4L Viewer permite cargar Topic Maps y visualizarlos gráficamente, así como mantener abiertos varios Topic Maps. El sistema permite al usuario cerrar el Topic Map que se está visualizando actualmente.

Mezclar Topic Map

La herramienta permite realizar mezclas de dos Topic Maps. Cuando se produce la mezcla de Topic Maps, algunos Topics que tienen el mismo *subject* son mezclados. Como consecuencia de la mezcla, algunas de las asociaciones duplicadas son borradas. El resultado final es un nuevo



Topic Map con la unión de las características de los dos Topic Maps unidos, pero con la información duplicada eliminada.

Elementos Topic Map

Después de cargar un Topic Map, se pueden visualizar los siguientes elementos:

- Subject Topics.
- Relationships.
- Topic Types.
- Relationship Types.
- Resource Types.
- Themes.

2.3.5.4 Dependencias

No se han encontrado dependencias importantes; simplemente se debe indicar que necesita máquina virtual de Java para poder funcionar correctamente.

2.3.5.5 Licencia

Es un software de libre distribución.

2.3.6. TopicMapDesigner

2.3.6.1 Descripción

Topic Map Designer [7] es una herramienta creada por el profesor Dr. Eric Schoop para su tesis doctoral, por lo que la herramienta no ha sido actualizada desde su construcción. El programa soporta ISO 13250 pero puede exportar un topic map formato XTM 1.0.

Una limitación de la herramienta es que no soporta *scopes*, lo que hace que las prestaciones de la herramienta sean menores.

Si se desea encontrar mayor información sobre esta herramienta, la misma está disponible en la dirección: <http://www.topicmap-design.com/en/topicmap-designer.htm>

2.3.6.2 Vistas del TopicMapDesigner

Designer

Esta vista permite al usuario ir seleccionado de la lista que aparece a la izquierda de la pantalla los componentes que desea visualizar. El funcionamiento es tan simple como arrastrar de la lista de topics y associations y soltar en el *designer view*.

Los topics son representados por cuadrados rojos y las associations por triángulos azules. Para eliminar topics o associations del *designer view*, simplemente hay que seleccionar el elemento que se desea eliminar de la vista y pulsar sobre la tecla de borrado. Esta acción borra el topic o la association de la vista, pero no del Topic Map.

Browser

Mediante esta vista, el usuario dispone de un navegador web que le permitirá navegar por Internet.

Viewer

Esta vista permite al usuario ver el topic map representado gráficamente. La representación se realiza mediante una especie de esfera, y en su interior se representan los topics y las asociaciones entre ellos. Topic Map Designer permite al usuario seleccionar el topic central de la esfera, mostrándose como centro del gráfico, siendo identificado por el color amarillo.

2.3.6.3 Funcionalidad del TopicMapDesigner

La herramienta permite añadir nuevos topics y associations. En un treeview que aparece a la izquierda se pueden encontrar todos los topics y las associations; seleccionando un elemento del treeview se mostrará toda la información del elemento seleccionado. Esta información del elemento puede ser modificada, añadiendo y suprimiendo topicnames y ocurrencias. En las associations además se pueden añadir y eliminar roles.

2.3.6.4 Dependencias

No se han encontrado dependencias importantes.



2.3.6.5 Licencia

Es un software de libre distribución.

2.3.7. Ontopia Omnigator

2.3.7.1 Descripción



Omnigator [8] es un sistema con una arquitectura cliente-servidor basado en el protocolo http. En la parte del servidor se aloja una aplicación web J2EE que corre bajo un servidor de aplicaciones Tomcat. Esta aplicación se encarga de leer Topic Map y generar páginas html con el resultado; el cliente simplemente tendrá que visualizar las páginas creadas por el servidor. La visualización está formada en casi su totalidad de links que permiten al usuario ir descubriendo información del Topic Map.

Si se desea encontrar mayor información sobre esta herramienta, la misma está disponible en la dirección: <http://www.ontopia.net/omnigator/models/index.jsp>

2.3.7.2 Modos de visualizar los Topic Map

Modo gráfico:

En el modo gráfico, aparecen los topics representados como nodos y las asociaciones como arcos. Las formas y los colores denotan diversos topic types y association types. Si se pasa el ratón por encima de uno de los arcos, se muestra el nombre del tipo de asociación.

Ontopia Omnigator está provista de un zoom que permite visualizar el gráfico de diferente tamaño, y gracias a un scroll bar nos permite movernos por todo el gráfico sin complicación alguna. El visualizador permite arrastrar los nodos de manera que el usuario puede colocar los topics del modo que le resulte más sencillo.

Las características de cada Topic pueden visualizarse pulsando con el botón derecho del ratón sobre el topic y seleccionando *Properties*.

Otra funcionalidad del modo gráfico, es que permite añadir o eliminar topics en la vista. Esta operación puede realizarse de tres formas distintas:

1. Aumentando o disminuyendo el *Locality factor*; es decir, aumentando o disminuyendo el número que aparece a la izquierda de la barra del zoom. El problema de este método es que el tiempo que tarda en realizar el proceso es bastante elevado.
2. Haciendo clic en algún nodo que tenga una pequeña marca roja en la esquina derecha superior. Esta marca significa que el topic tiene asociaciones adicionales que no se están mostrando (si se hace clic, muestra esas asociaciones).
3. Mediante la funcionalidad que permite *Expand node*, *Collapse node* o *Hide node*. El *Expand node* tiene el mismo efecto que pulsando en el nodo; el *Collapse node* tiene el efecto contrario que el *Expand node*; el *Hide node* oculta el nodo completamente.

Modo texto:

En el modo texto, el sistema lo que hace es mostrarte la información de todos los topics en cuadros con enlaces que permiten navegar por la información del TM. Un problema que se ha

encontrado es que cuando las listas son muy largas quedan truncadas, por lo que no se puede visualizar toda la información completa.

2.3.7.3 Funcionalidad Ontopia Omnigator

Abrir y cerrar un Topic Map

Ontopia Omnigator permite cargar Topic Maps ya creados anteriormente y visualizarlos gráficamente.

Elementos Topic Map

Después de cargar un Topic Map, se pueden visualizar los siguientes elementos:

- Metadatos del Topic Map
- Topic Types
- Association Types
- Association rol Types
- Occurrence Types
- Topics
 - Names
 - Associations
 - Occurrences
- Associations
 - Names
 - Role Type
- Themes
 - Base name Themes
 - Occurrence themes

Plugins del sistema

Ontopia Omnigator, además de ser una herramienta que permite visualizar Topics Maps, dispone de una serie de plugins adicionales que permiten ampliar su funcionalidad. Los plugins más destacados son:

The edit plug-in

El plug-in dota a la herramienta de la capacidad de editar la información de los Topic Maps. El principal problema del plug-in es que trabaja sobre unos Topic Maps especiales, a los que se ha



denominado Ontopoly Topic Map. Si se desea editar un Topic Map que no ha sido creado con la herramienta, el sistema avisará que se debe actualizar el Topic Map actual o se debe crear uno nuevo.

Filtro

Este plug-in permite modificar el scope del Topic Map, estableciendo un contexto dentro del cual las características del topic son válidas.

Merge plug-in

El plug-in permite mezclar un Segundo Topic Map con el actual. La mezcla de los dos Topic Maps se realizará eliminando información duplicada.

Export plug-in

La finalidad de este plug-in es poder exportar un Topic Map en distintos formatos. Los formatos que soportan son: XTM, LTM, TM/XML y RDF.

Query plug-in

El Query plug-in permite al usuario realizar búsquedas en los Topic Maps usando un lenguaje llamado *tolog*. Tolog es un query language creado para realizar búsquedas en Topic Maps, desarrollado por Ontopia. A continuación se expondrá un pequeño ejemplo de uso de dicho lenguaje:

Ejemplo de Tolog

```
select $COMPOSER, count($OPERA) from
composed-by($OPERA : opera, $COMPOSER : composer)
order by $OPERA desc?
```

Esta query busca todas las óperas y compositores, y devuelve una lista de compositores y el número de óperas que compusieron, clasificados en orden descendente.

En la siguiente página web (<http://www.ontopia.net/omnigator/docs/query/tutorial.html>) se encuentra un manual del uso de este lenguaje.

Statistics plug-in

Este plug-in crea una serie de estadísticas sobre el Topic Map. Proporciona una descripción de la estadística “vital” del Topic Map y de una estadística detallada de alguno de sus elementos como topics, asociaciones, etc. Esta información puede ser de utilidad para revelar inconsistencias u otros problemas del Topic Map.

2.3.7.4 Dependencias

No se han encontrado dependencias importantes; simplemente se debe indicar que necesita máquina virtual de Java para poder funcionar correctamente.



2.3.7.5 Licencia

El funcionamiento de la actual implementación de Ontopia Omnigator depende de Apache, ya que es una estructura cliente servidor. Además necesita una máquina para alojar el servidor, la cual puede disponer de distintos sistemas operativos tales como Windows, Linux o Mac OS X.

Ontopia Omnigator es una aplicación cliente servidor, lo que implica que depende del protocolo http y, por consiguiente, el cliente podrá comunicar con el servidor siempre y cuando el protocolo http no sufra cambios importantes.



2.3.8. Conclusiones

El mercado actual dispone de una serie de herramientas que permiten editar y visualizar Topic Maps. El estudio realizado anteriormente de las herramientas pone en evidencia que la mayoría de los editores estudiados no explotan todo el potencial del estándar de Topic Map.

El editor más destacado de todos los analizados previamente es TM4L-Editor. Este editor permite al usuario aprovechar todo el potencial del estándar de Topic Map de una forma más o menos sencilla, aunque en ocasiones se producen errores que no están solucionados, y que dificultan la creación del Topic Map.

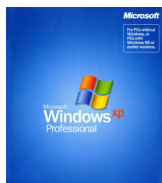
De los visualizadores de Topic Map estudiados, el más destacado es TM4L-Viewer ya que dispone de varios modos de visualización que permiten al usuario estudiar el Topic Map de distintas formas. Hay que destacar la dificultad de todas las herramientas analizadas para mostrar Topic Maps demasiado grandes, por el gran número de elementos que lo componen.

3. Herramientas

En este punto se expondrán todas las herramientas, tanto software como hardware, que permitirán realizar este proyecto.

3.1. Software

A continuación, se presentará una breve descripción del software necesario para la creación del sistema.



Sistema Operativo – Windows XP Profesional

Si bien siempre nos podríamos decantar por software libre, no podemos obviar la enorme cantidad de productos que existen para el desarrollo software bajo un entorno Windows; es por ello por lo que la máquina de desarrollo incluye el más extendido de los sistemas operativos en el mundo, en su versión profesional.

A continuación se describirá el software básico a utilizar, tanto en el proceso de desarrollo propiamente dicho como en la implantación del sistema.

3.1.1. Software de desarrollo

El software de desarrollo que se va a necesitar es el siguiente:



Microsoft Office 2007

Se utilizará la suite por excelencia de Microsoft en los estándares de documentación, ya que es la herramienta más usada en el mercado actual; además, el paquete ofrece todas las funcionalidades necesarias para realizar todo el trabajo de documentación del proyecto.



Microsoft Project 2007

El proceso de planificación es realmente importante; por ello, se utilizará la herramienta más difundida y completa del mercado. Dicha herramienta permite programar y organizar los recursos y tareas, a fin de generar proyectos a tiempo y conforme a su presupuesto.



Entornos de desarrollo Eclipse

Para el proceso de desarrollo se dispondrá un entorno de libre distribución, en los que colaboran las empresas más punteras del mercado. Este entorno nos proporciona la máxima potencia a la hora de construir nuestro software.

3.1.2. Software de implantación

La creación del software a desarrollar se realizará bajo el siguiente entorno tecnológico, tanto de paquetes software como especificaciones, que serán descritas a continuación.



Java

El lenguaje de programación Java fue diseñado por la compañía Sun Microsystems Inc, con el objetivo de crear un lenguaje que pudiera funcionar en redes computacionales heterogéneas, y que fuera independiente de la plataforma en la que se vaya a ejecutar. Esto significa que un programa de Java puede ejecutarse en cualquier máquina o plataforma.

La elección de Java como lenguaje de programación se debe a una serie de razones:

- *Simple*: Elimina la complejidad de lenguajes tales como "C" y da paso al contexto de los lenguajes modernos orientados a objetos.
- *Familiar*: Como la mayoría de los programadores están acostumbrados a programar en C o en C++, la sintaxis de Java es muy similar al de éstos.
- *Robusto*: Java no sólo verifica el código para encontrar problemas en tiempo de compilación, sino que también busca errores que se puedan presentar en tiempo de ejecución.
- *Seguro*: Debido a que Java fue diseñado para correr en un ambiente de red, existen restricciones de seguridad. Además, durante la ejecución, el intérprete utiliza un mecanismo para verificar que el *byte code* cargado a través de la red no viola ninguna restricción del lenguaje Java.
- *Portable*: Como el código compilado de Java (conocido como *byte code*) es interpretado, un programa compilado de Java puede ser utilizado por ordenador que tenga implementado el intérprete de Java.
- *Independiente a la arquitectura*: Al compilar un programa en Java, el código resultante es un tipo de código binario conocido como *byte code*. Dicho código es interpretado por diferentes máquinas de igual manera; solamente hay que implementar un intérprete para cada plataforma. De esta manera, Java logra ser un lenguaje que no depende de una arquitectura computacional definida.
- *Interpretado*: Java corre en máquina virtual, por lo tanto es interpretado.
- *Dinámico*: Java no requiere la compilación de todas las clases de un programa para que éste funcione. Si se efectúa una modificación a una clase, Java se encarga de realizar un *Dynamic Binding* o un *Dynamic Loading* para encontrar las clases.

3.2. Hardware

Si bien el software con el que se desarrolla la aplicación, así como el hardware sobre el que se implanta, son importantes, no lo es menos el hardware sobre el que se desarrollará. A continuación, expondremos una breve descripción de las tecnologías hardware sobre las que se desarrollará el sistema.

3.2.1. Hardware de desarrollo

Para el desarrollo del sistema se utilizarán dos máquinas con las siguientes prestaciones:



Proyecto Fin de Carrera-TMCreator

- Equipo portátil Acer Travelmate 4152LMI 1.7 Ghz, 1 GByte de RAM
- Equipo de sobremesa PC 1,7 Ghz, 1 Gbyte de RAM para realizar copias de seguridad.
- Acceso a Internet mediante un router Conceptronic C54APRA.

4. Análisis del sistema

4.1. Requisitos de usuario

El catálogo de requisitos de usuario especificará todos y cada uno de los requisitos de usuario del proyecto TMCreator.

Se utilizará una tabla como la siguiente para la catalogación de requisitos:

RUX-NN	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	Y.Y
DESCRIPCIÓN:	Descripción del requisito.

Tabla 2. Modelo de especificación de un requisito de usuario

En dicha tabla, se describirán los siguientes atributos:

- **Identificación:** cada requisito incluirá una identificación que será única para cada uno de ellos. Se nombran con las siglas RUX (requisito de usuario de X) seguidos de un guión y dos dígitos enumerados de manera secuencial. La X podrá ser C si es un requisito de capacidad o R si es un requisito de restricción. NN seguirá una numeración secuencial, empezando por 01 hasta 99.
- **Prioridad:** hace referencia al orden temporal en el que debe realizarse ese requisito. Los valores que puede tomar este campo son: prioridad alta, media y baja.
- **Estabilidad:** señala si un requisito es o no estable. Un requisito *no estable* es aquel que está sujeto a posibles modificaciones, ya sea por el propio cliente o bien por otros factores que puedan afectar al sistema. Los valores de este atributo son sí o no.
- **Claridad:** muestra la facilidad o dificultad para aplicar el requisito en el sistema. Normalmente está relacionada con la dificultad para comprender el significado del requisito.
- **Verificabilidad:** es un indicador de la dificultad o facilidad para comprobar si el requisito se cumple en el sistema final.



- **Necesidad:** indica si un requisito es esencial o no. Se utilizará una escala de tres niveles: esencial, deseable y opcional.
- **Versión:** permite visualizar la traza de los distintos estados por los que haya pasado un requisito para identificarlo unívocamente.

4.1.1. Requisitos de capacidad

Los requisitos de capacidad describirán todas las funcionalidades disponibles en el sistema TMCreator.

RUC-01	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá crear un nuevo Topic Map.

RUC-02	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá grabar el Topic Map en un fichero con formato xtm.

RUC-03	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO



Proyecto Fin de Carrera-TMCreator

RUC-03	
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá cargar un Topic Map creado anteriormente en formato xtm.

RUC-04	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá crear nuevos Topics.

RUC-05	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá eliminar los Topics que considere oportunos.



Proyecto Fin de Carrera-TMCreator

RUC-06	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá modificar la información que almacena cada Topic.

RUC-07	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	EL usuario podrá completar la información de cada topic indicando: <ul style="list-style-type: none">• Topic Name• Subject• Subject indicador• Occurrences del Topic• Base Names• De 0 a n Topics Resources

RUC-08	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0



Proyecto Fin de Carrera-TMCreator

RUC-08	
DESCRIPCIÓN:	El usuario podrá crear asociaciones entre Topics.

RUC-09	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá eliminar las asociaciones que considere oportunas.

RUC-10	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá modificar la información que almacena cada asociación.

RUC-11	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA



RUC-11	
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá indicar: <ul style="list-style-type: none">• Nombre de la asociación• Roles de los miembros de la asociación• Scope de la asociación• Los miembros de la asociación

RUC-12	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá visualizar los topics que estén ligados al scope seleccionado.

RUC-13	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	Al crear un nuevo scope, el usuario podrá indicar: <ul style="list-style-type: none">• Nombre del scope• Url del scope

RUC-14



RUC-14	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El sistema debe ser capaz de soportar metadatos.

4.1.2. Requisitos de restricción

RUR-01	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El sistema deberá seguir el estándar xtm en su versión 1.0.

4.2. Modelo de casos de uso

Casos de uso:

Todo caso de uso está identificado unívocamente por un código que se describe a continuación:

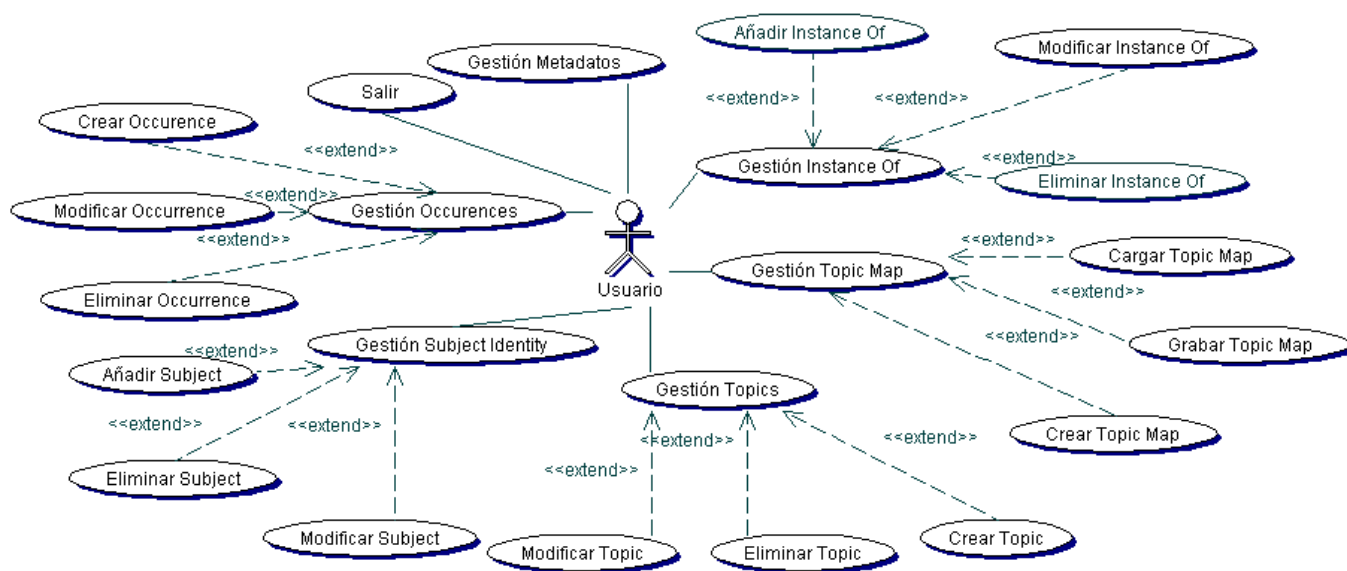
- Los dos primeros caracteres (CU) son comunes a todos los requisitos y, correspondientes al acrónimo Caso de Uso.
- A continuación, 4 dígitos que identifican al caso de uso. Entre los 2 primeros dígitos y los siguientes se colocará un “-“.
 - 01:Gestión Topic Map
 - 1:Crear Topic Map
 - 2:Cargar Topic Map
 - 3:Grabar Topic Map
 - 02:Gestión Topics
 - 1:Crear Topic
 - 2:Eliminar Topic
 - 3:Modificar Topic
 - 03:Gestión Occurrences
 - 1:Crear Occurrence
 - 2:Eliminar Occurrence
 - 3:Modificar Occurrence
 - 04:Gestión Base Names
 - 1:Añadir Base Name
 - 2:Eliminar Base Name
 - 3:Modificar Base Name
 - 05:Gestión Asociaciones
 - 1:Crear Asociación
 - 2:Eliminar Asociación
 - 3:Modificar Asociación
 - 06:Gestión Roles
 - 1:Crear Rol
 - 2:Eliminar Rol
 - 3:Modificar Rol
 - 07:Gestión Members
 - 1:Añadir Member
 - 2:Eliminar Member
 - 3:Modificar Member
 - 08:Gestión Scopes
 - 1:Visualizar elemento Scope
 - 09:Gestión instance Of
 - 1:Añadir Instance Of
 - 2:Eliminar Instance Of
 - 3:Modificar Instance Of
 - 10:Gestión Subject Identity

- 1:Añadir Subject Identity
- 2:Eliminar Subject Identity
- 3:Modificar Subject Identity
- 11:Salir
- 12:Gestión de Metadatos

Diagramas de casos de uso:

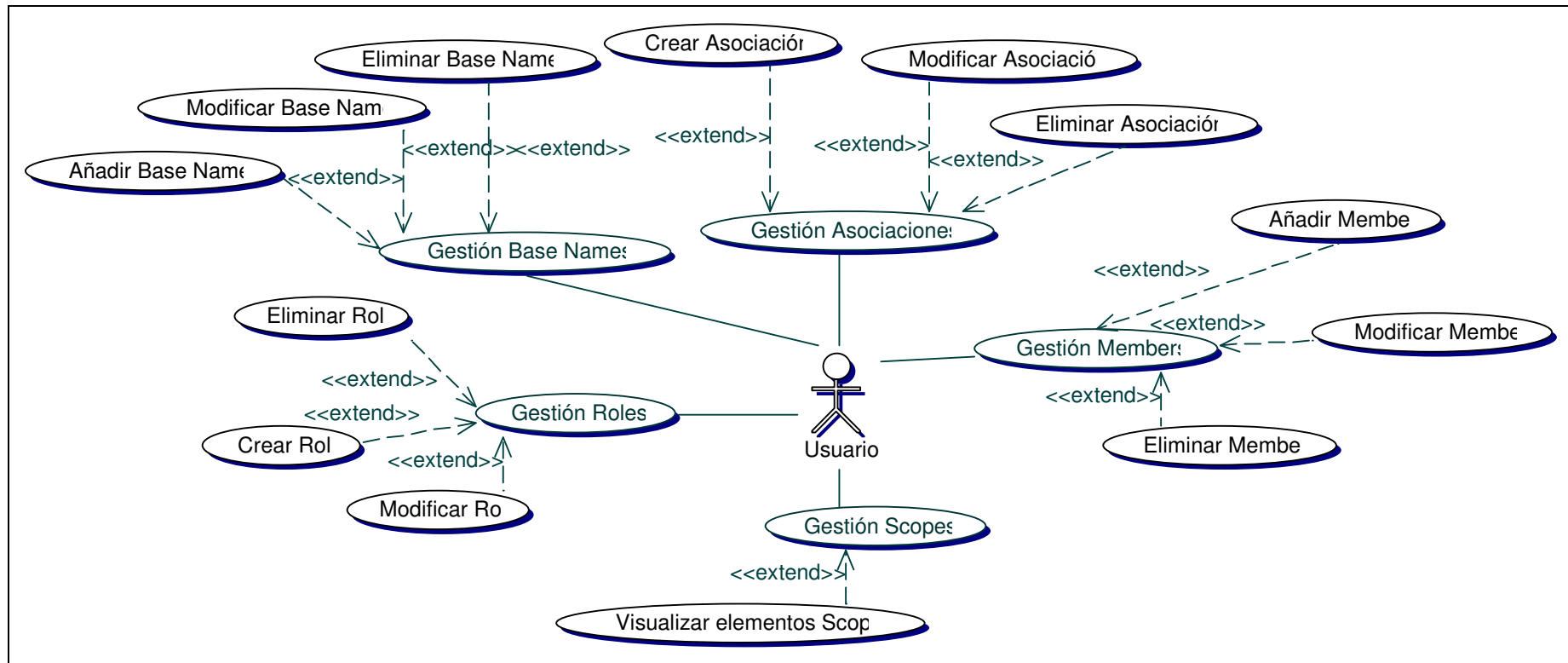
El diagrama de casos de uso lo voy a separar en dos partes para una mejor visualización.

Escenario del Usuario (I):





Escenario del Usuario (II):



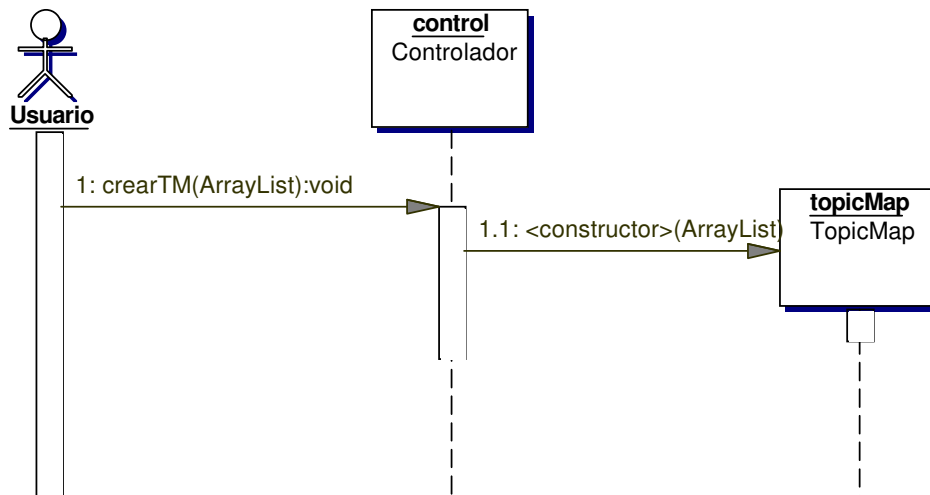
Descripción de cada caso de uso:

CU01: Gestión Topic Maps

CU01-1: Crear Topic Map

Caso de Uso:	Crear Topic Map
Objetivo:	Crear un nuevo Topic Map
Actores:	Usuario(U)
Precondiciones:	-
Postcondiciones:	Se ha creado un nuevo TM
Escenario básico:	1.U: el usuario pulsa sobre el botón que permite crear un nuevo TM 2.S: pide al usuario la información necesaria para crear un nuevo TM 3.S: crea un nuevo TM

Diagrama de secuencia del caso de uso





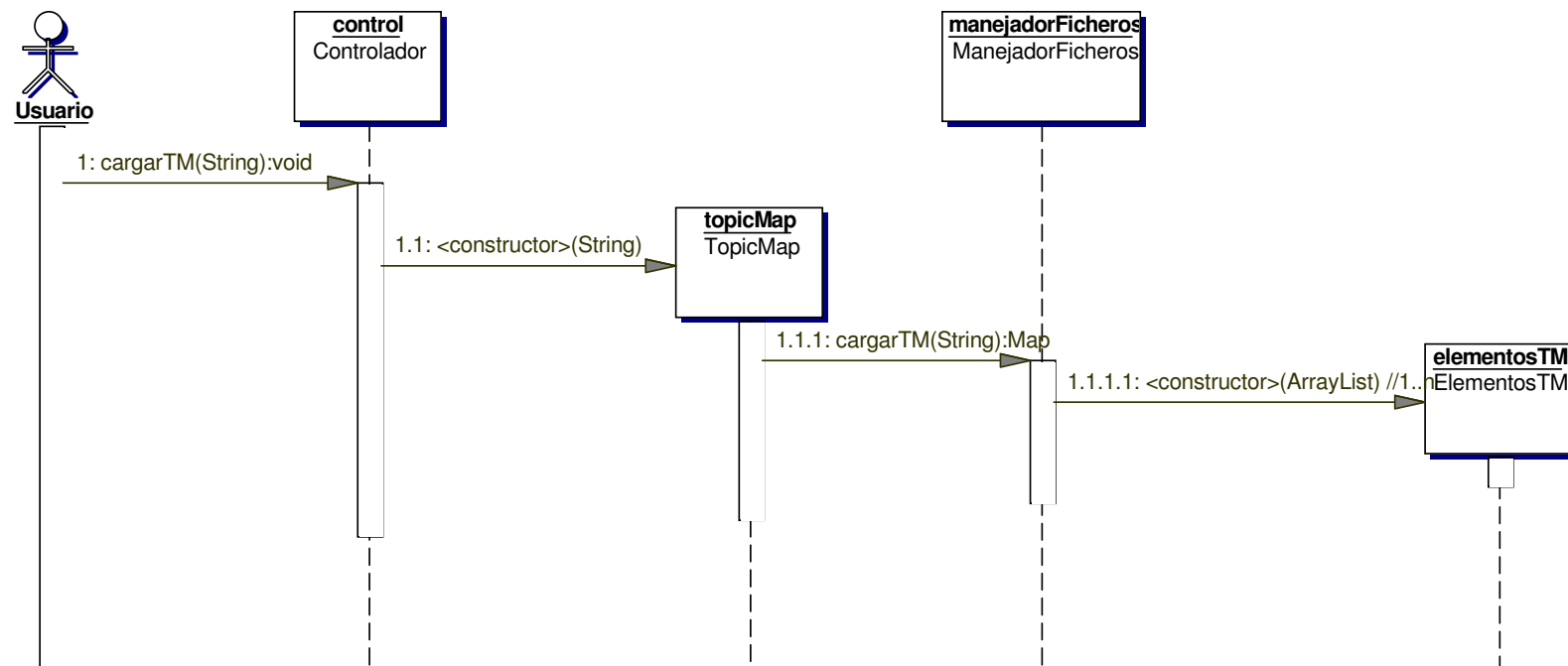
CU01: Gestión Topic Maps

CU01-2: Cargar Topic Map

Caso de Uso:	Cargar Topic Map
Objetivo:	Cargar un TM a partir de un fichero xtm
Actores:	Usuario(U)
Precondiciones:	-
Postcondiciones:	
Escenario básico:	<ul style="list-style-type: none">1.U: El usuario inicia la carga del fichero xtm2.S: El sistema lee el fichero indicado por el usuario2.1 S: Si el fichero no existe o no está bien formado, el sistema indica al usuario que el fichero no es válido3.S: Si el fichero es válido, el sistema carga el fichero



Diagrama de secuencia del caso de uso

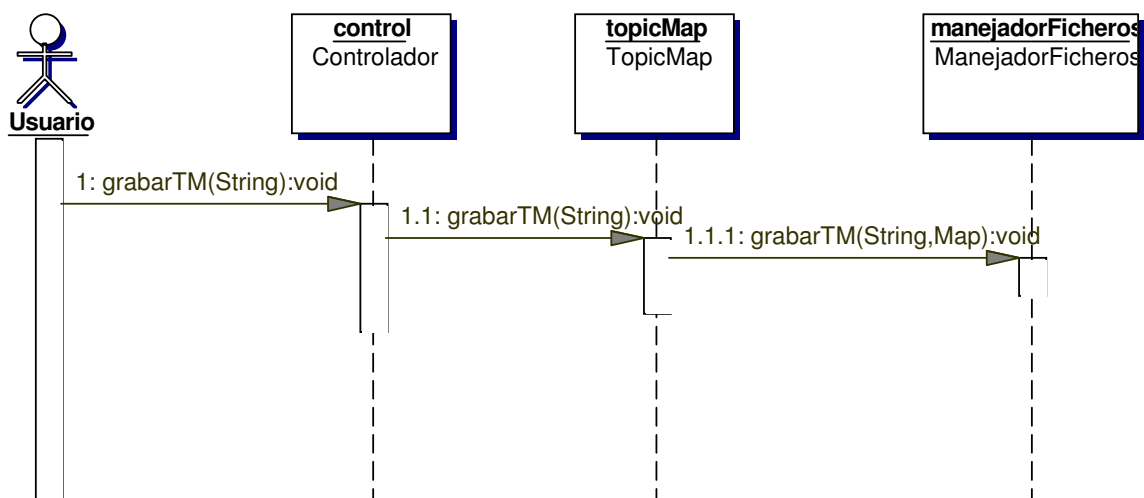


CU01: Gestión Topic Maps

CU01-3: Grabar Topic Map

Caso de Uso:	Grabar Topic Map
Objetivo:	Almacenar en el disco duro el Topic Map que hemos creado en formato xtm.
Actores:	Usuario(U)
Precondiciones:	El usuario debe haber creado un Topic Map
Postcondiciones:	El Topic Map quedará guardado en un fichero con extensión xtm
Escenario básico:	<ol style="list-style-type: none">1.U: el usuario inicia el proceso de guardado y selecciona un nombre para el fichero2.S: el sistema crea un nuevo fichero con el nombre dado por el usuario y almacena en él toda la información del TM

Diagrama de secuencia del caso de uso

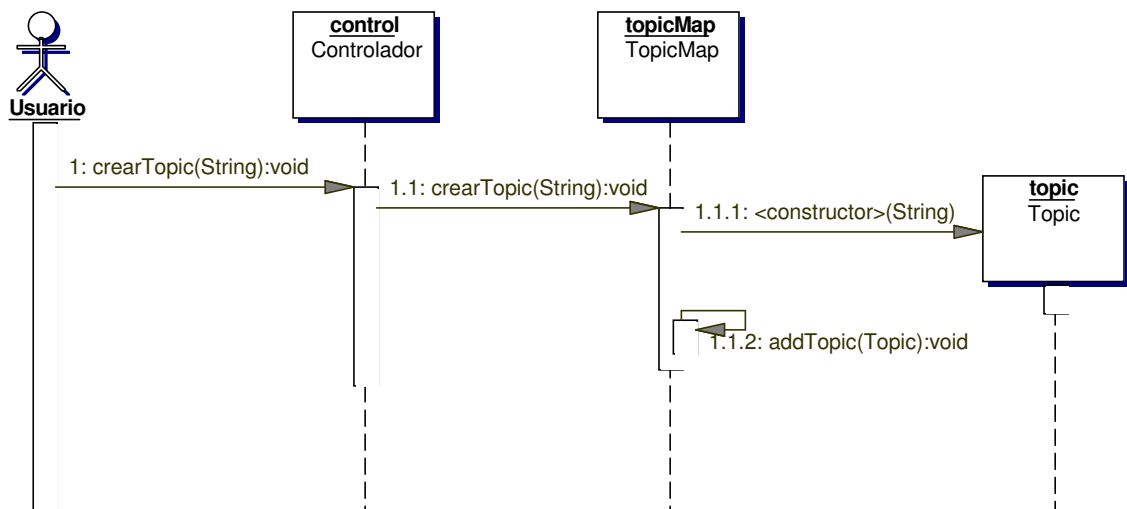


CU02: Gestión Topics

CU02-1: Crear Topic

Caso de Uso:	Crear Topic
Objetivo:	Crear un nuevo Topic
Actores:	Usuario(U)
Precondiciones:	Que exista un Topic Map en el que se creará el Topic
Postcondiciones:	Se habrá creado un nuevo Topic
Escenario básico:	<ol style="list-style-type: none">1.U: el usuario introduce el nombre del Topic que desea crear2.S: comprueba que el nombre no existe como Topic en el Topic Map2.1 S: si existe, muestra un mensaje de error; en caso contrario, continúa con el punto 33.S: crea el Topic

Diagrama de secuencia del caso de uso



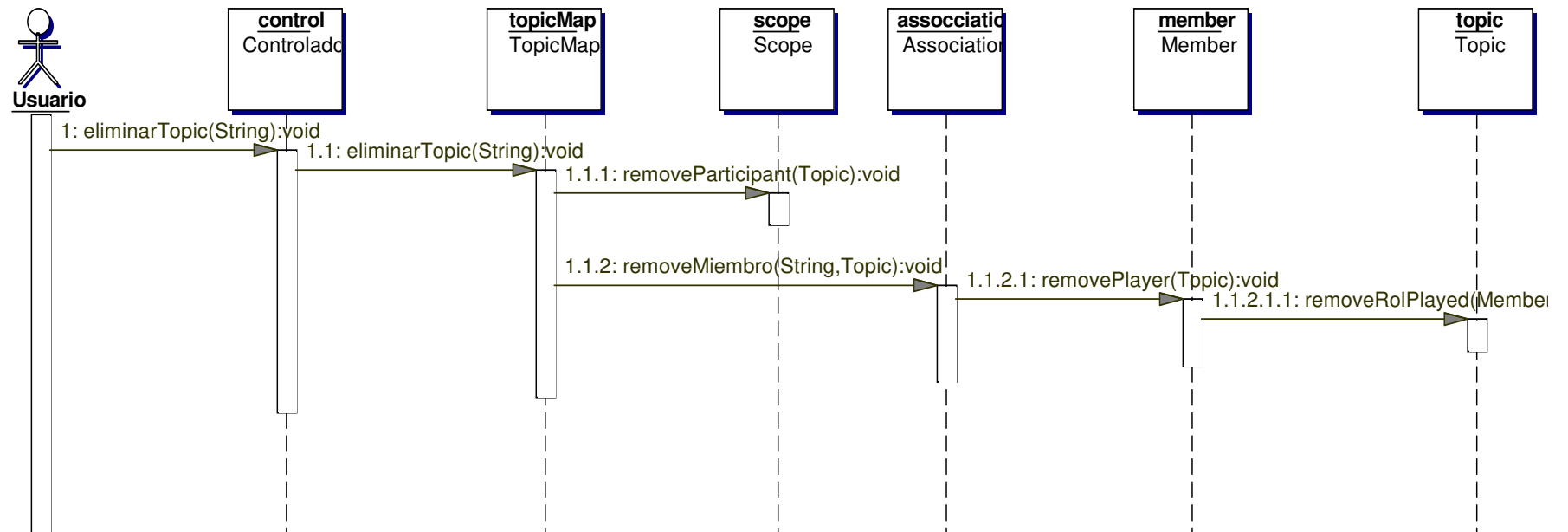


CU02: Gestión Topics

CU02-2: Eliminar Topic

Caso de Uso:	Eliminar Topic
Objetivo:	Eliminar un Topic seleccionado previamente
Actores:	Usuario(U)
Precondiciones:	Que exista el Topic que se desea eliminar
Postcondiciones:	Se habrá eliminado el Topic seleccionado
Escenario básico:	1.U: el usuario selecciona el Topic que desea eliminar 2.S: elimina el Topic del Topic Map

Diagrama de secuencia del caso de uso

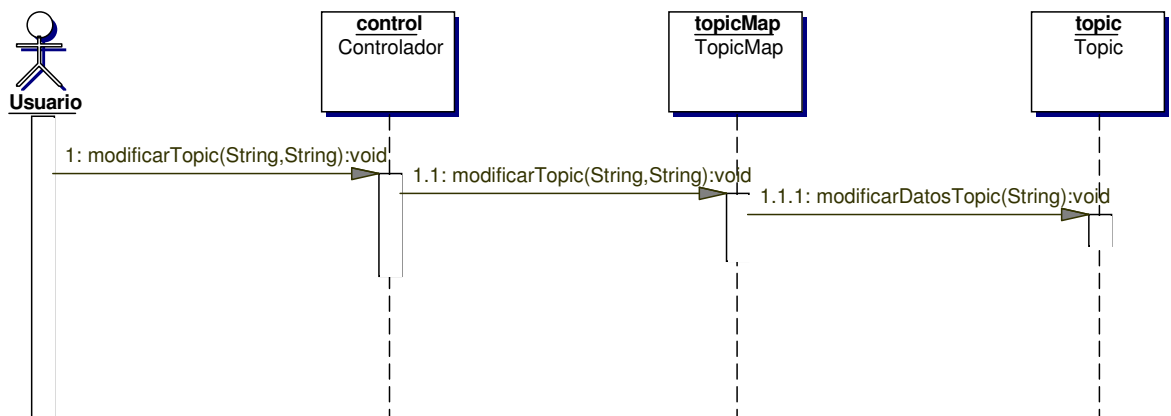


CU02: Gestión Topics

CU02-3: Modificar Topic

Caso de Uso:	Modificar Topic
Objetivo:	Modificar la información almacenada por el Topic
Actores:	Usuario(U)
Precondiciones:	El Topic debe existir
Postcondiciones:	Los datos del Topic serán modificados
Escenario básico:	<ol style="list-style-type: none">1.U: introduce los nuevos datos2.S: comprueba que los datos son correctos<ol style="list-style-type: none">2.1 S: si no son correctos, muestra un mensaje; en caso contrario, continúa con el punto 33.S: modifica la información almacenada del Topic

Diagrama de secuencia del caso de uso



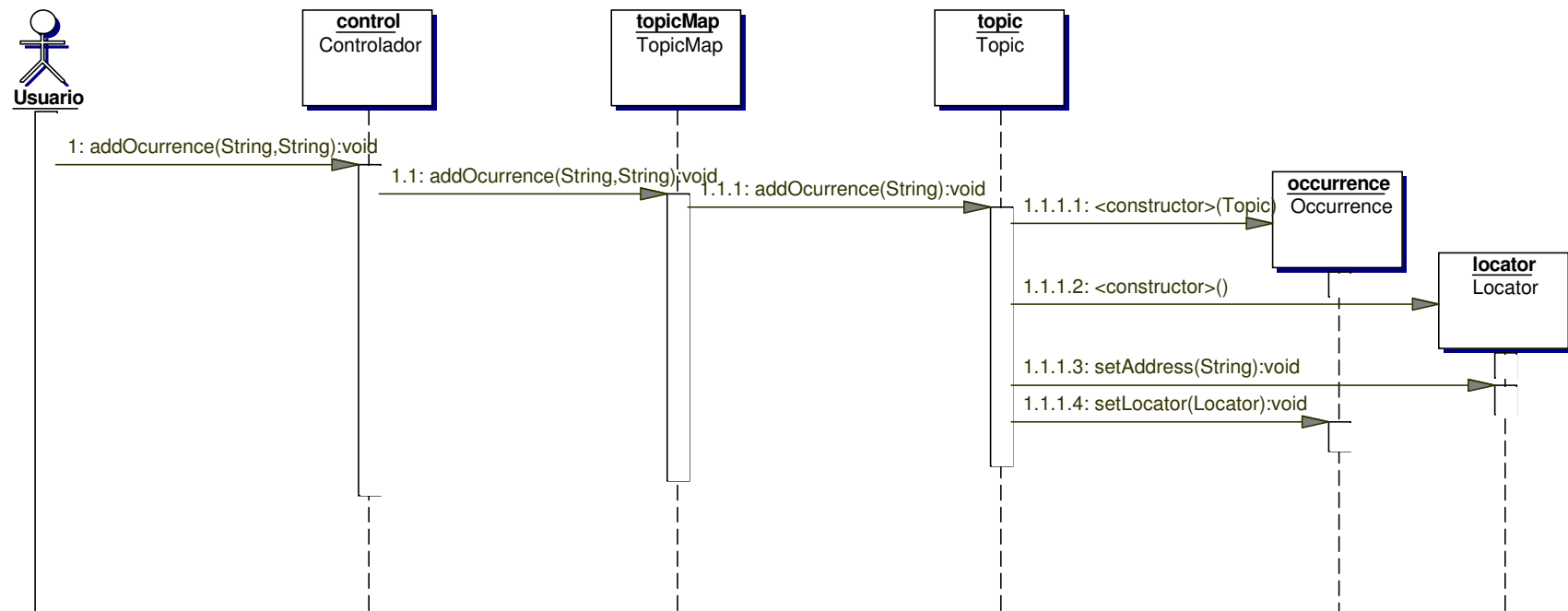


CU03: Gestión Occurrences

CU03-1: Crear Occurrence

Caso de Uso:	Crear Occurrence
Objetivo:	Crear una nueva Occurrence para el Topic seleccionado
Actores:	Usuario(U)
Precondiciones:	El Topic debe existir
Postcondiciones:	Se creará un nuevo Occurrence asociado al Topic
Escenario básico:	1.U: selecciona el Topic en el que desea crear la Occurrence e introduce la información de la Occurrence 2.S: crea la Occurrence asociada al Topic

Diagrama de secuencia del caso de uso

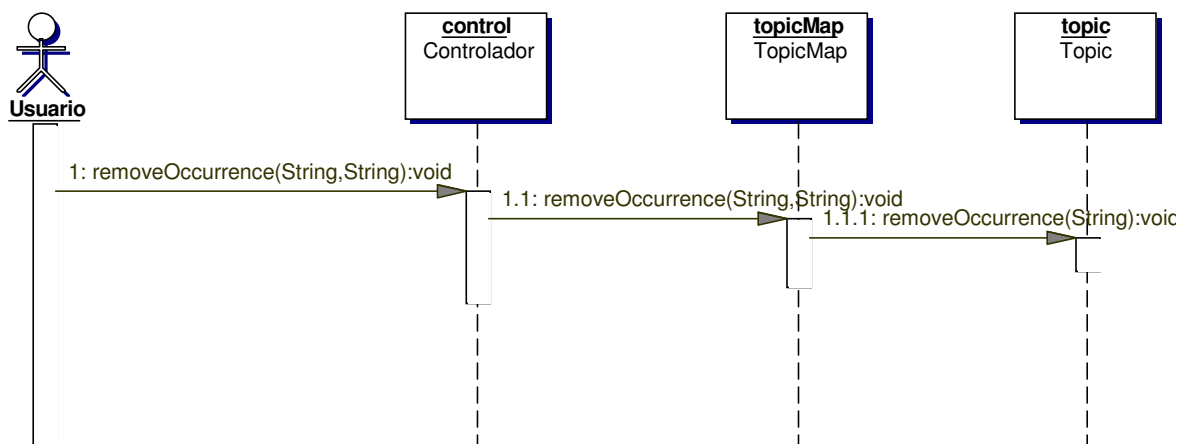


CU03: Gestión Occurrences

CU03-2: Eliminar Occurrence

Caso de Uso: Eliminar Occurrence
Objetivo: Eliminar la Occurrence seleccionada de un Topic cualquiera
Actores: Usuario(U)
Precondiciones: El Topic y la Occurrence deben existir
Postcondiciones: Se habrá borrado la Occurrence del Topic seleccionado
Escenario básico:
1.U: selecciona el Topic y después la Occurrence que desea eliminar
2.S: elimina la Occurrence asociada al Topic

Diagrama de secuencia del caso de uso



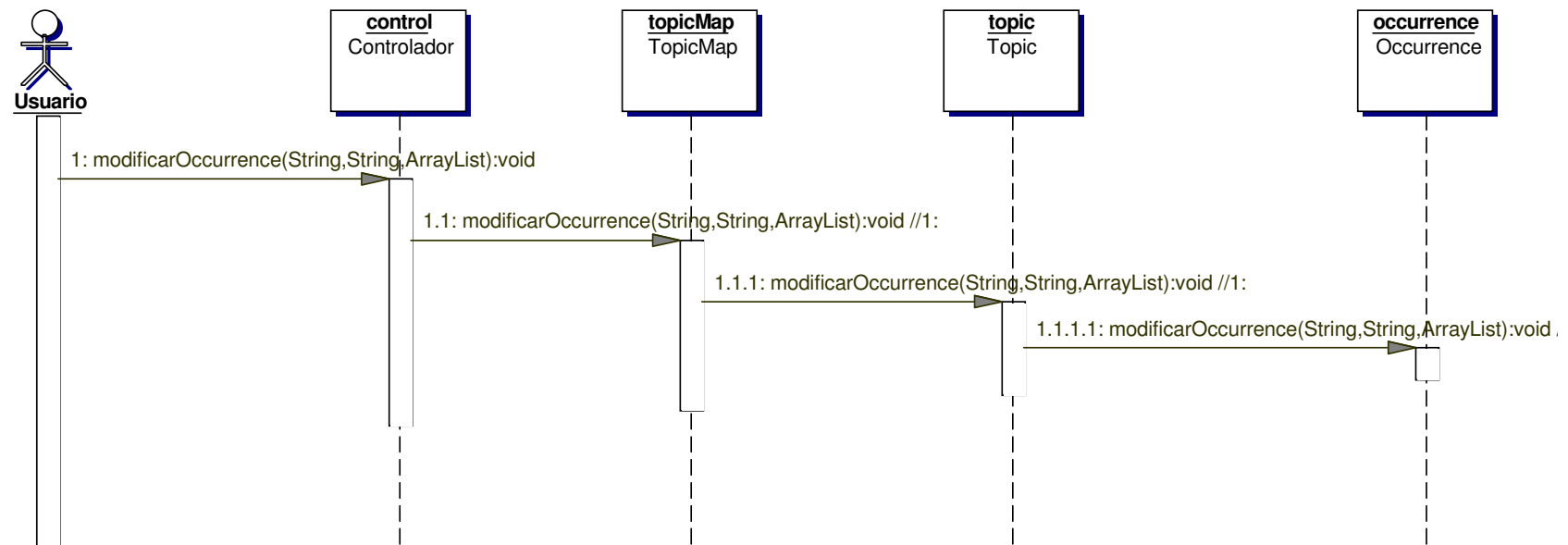


CU03: Gestión Occurrences

CU03-3: Modificar Occurrence

Caso de Uso:	Modificar Occurrence
Objetivo:	Modificar la Occurrence seleccionada de un Topic cualquiera
Actores:	Usuario(U)
Precondiciones:	El Topic y la Occurrence deben existir
Postcondiciones:	Se habrá modificado la Occurrence del Topic seleccionado
Escenario básico:	1.U: selecciona el Topic y después la Occurrence que desea modificar 2.S: modifica la Occurrence asociada al Topic

Diagrama de secuencia del caso de uso



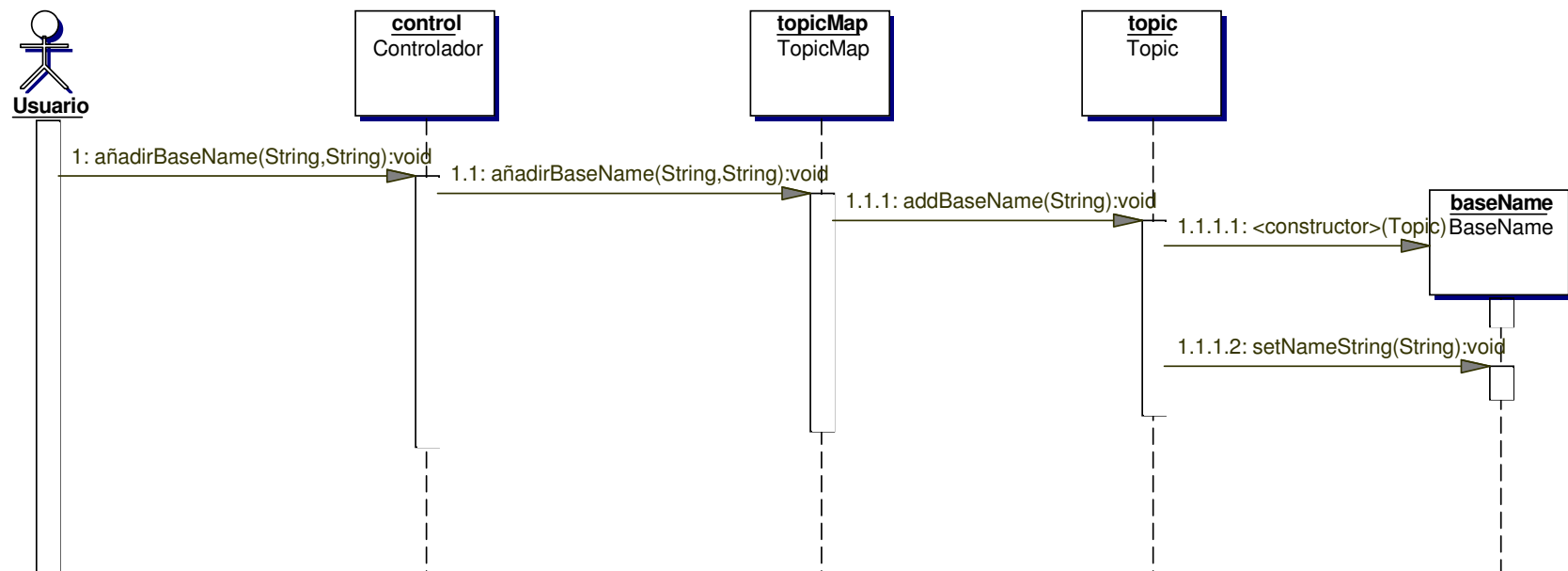


CU04: Gestión Base Names

CU04-1: Añadir Base Name

Caso de Uso:	Añadir Base Name
Objetivo:	Crear un nuevo Base Name en un Topic seleccionado
Actores:	Usuario(U)
Precondiciones:	El Topic debe existir
Postcondiciones:	Se creará un nuevo Base Name
Escenario básico:	1.U: selecciona el Topic en el que desea crear el Base Name e introduce la información necesaria 2.S: crea el Base Name asociado al Topic

Diagrama de secuencia del caso de uso



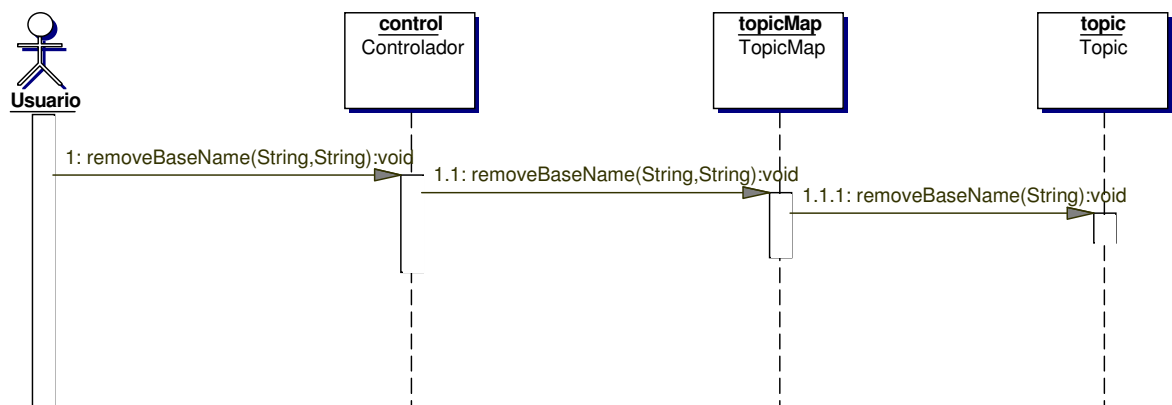


CU04: Gestión Base Names

CU04-2: Eliminar Base Name

Caso de Uso: Eliminar Base Name
Objetivo: Eliminar el Base Name seleccionado asociado a un Topic
Actores: Usuario(U)
Precondiciones: El Topic y el Base Name deben existir
Postcondiciones: Se habrá borrado el Base Name del Topic seleccionado
Escenario básico:
1.U: selecciona el Topic y después el Base Name que desea eliminar
2.S: elimina el Base Name asociado al Topic

Diagrama de secuencia del caso de uso



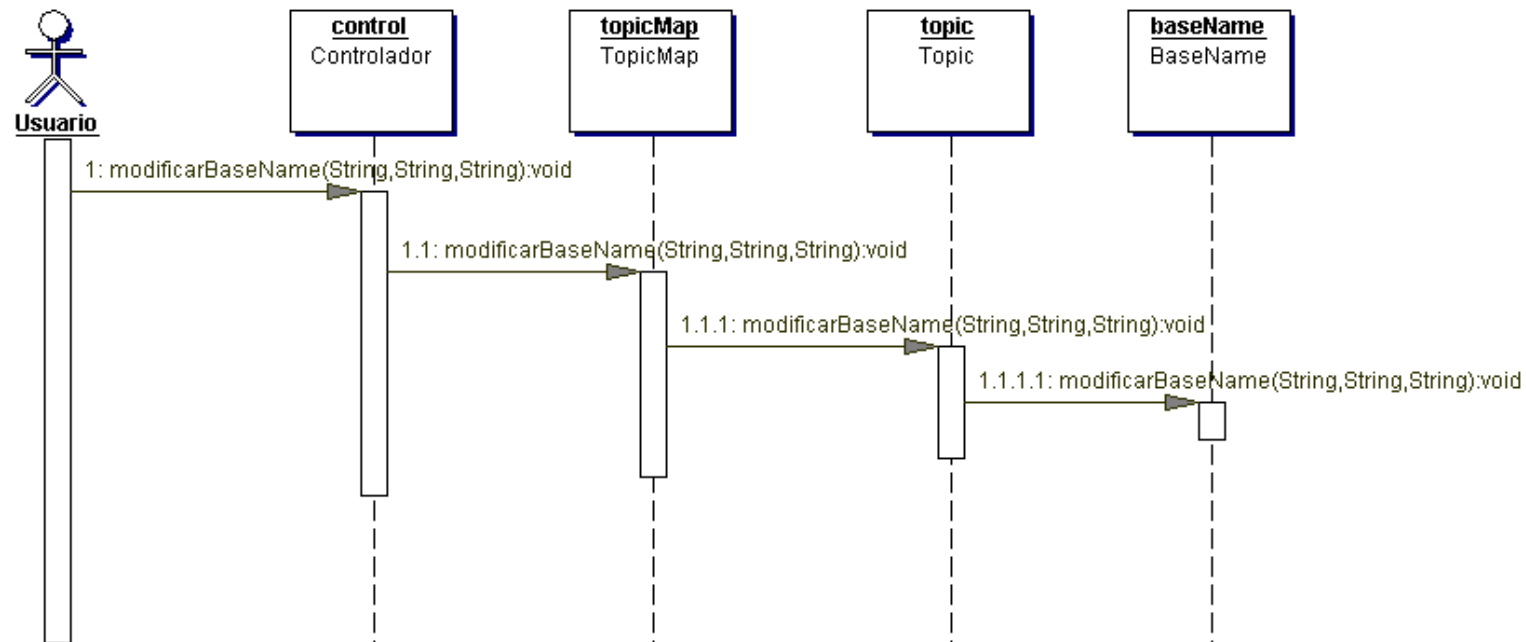


CU04: Gestión Base Names

CU04-3 Modificar Base Name

Caso de Uso:	Modificar Base Name
Objetivo:	Modificar el Base Name seleccionado asociado a un Topic
Actores:	Usuario(U)
Precondiciones:	El Topic y el Base Name deben existir
Postcondiciones:	Se habrá Modificar el Base Name del Topic seleccionado
Escenario básico:	1.U: selecciona el Topic y después el Base Name que desea modificar 2.S: modifica el Base Name asociado al Topic

Diagrama de secuencia del caso de uso

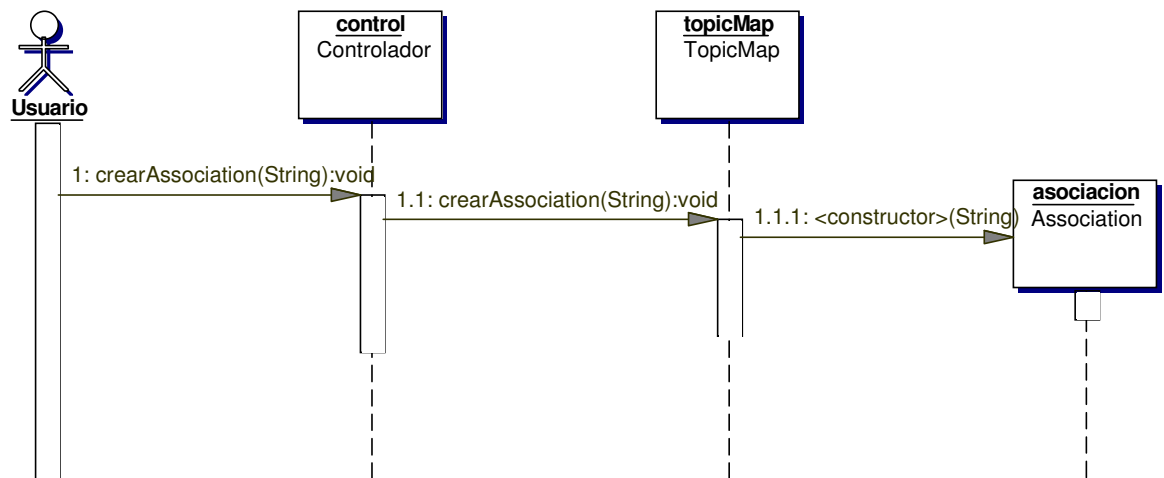


CU05: Gestión Asociaciones

CU05-1: Crear Asociación

Caso de Uso:	Crear Asociación
Objetivo:	Crear una nueva asociación
Actores:	Usuario(U)
Precondiciones:	Se haya creado un TM en el que ubicar la asociación
Postcondiciones:	Se habrá creado una nueva asociación
Escenario básico:	<ol style="list-style-type: none">1.U: el usuario introduce el nombre de la asociación2.S: el sistema comprueba que no exista una asociación con ese mismo nombre<ol style="list-style-type: none">2.1S: si existe, muestra un mensaje de error; en caso contrario, continúa con el punto 3.3.S: crea la asociación

Diagrama de secuencia del caso de uso



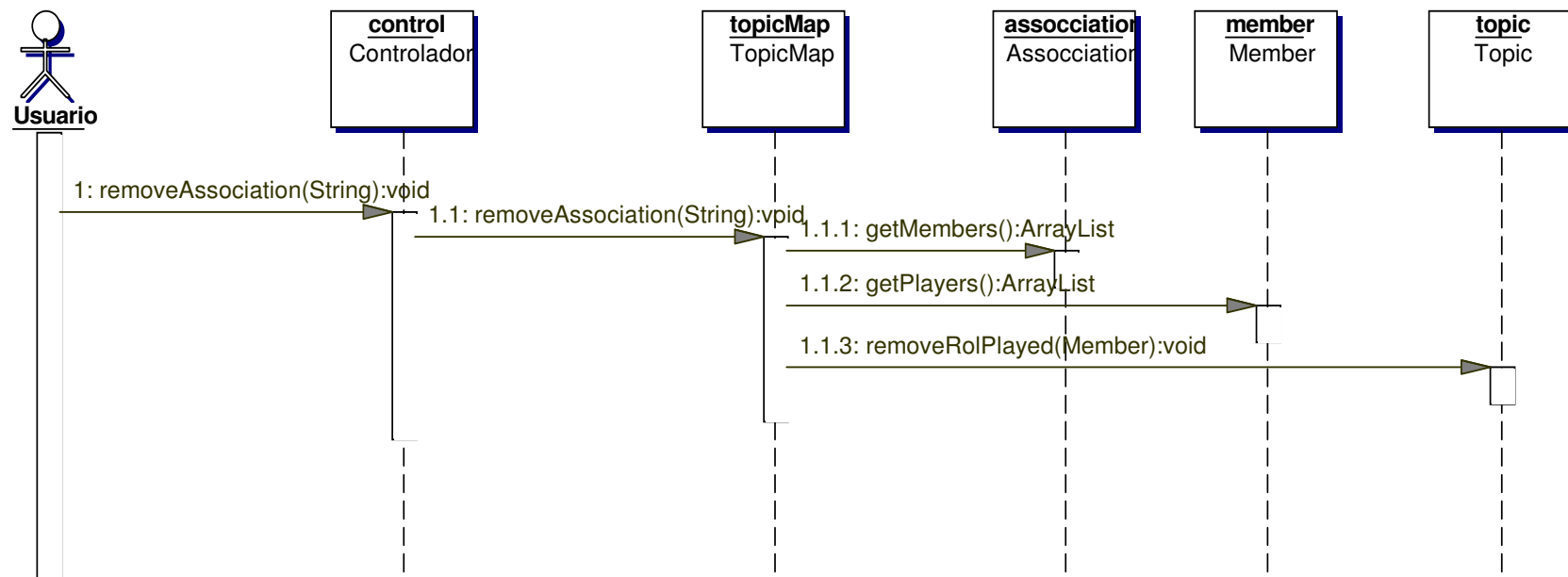


CU05: Gestión Asociaciones

CU05-2: Eliminar Asociación

Caso de Uso:	Eliminar Asociación
Objetivo:	Eliminar una asociación del Topic Map
Actores:	Usuario(U)
Precondiciones:	La asociación debe existir
Postcondiciones:	Se habrá borrado la asociación del Topic Map
Escenario básico:	1.U: selecciona la asociación que desea borrar 2.S: elimina la asociación del Topic Map

Diagrama de secuencia del caso de uso



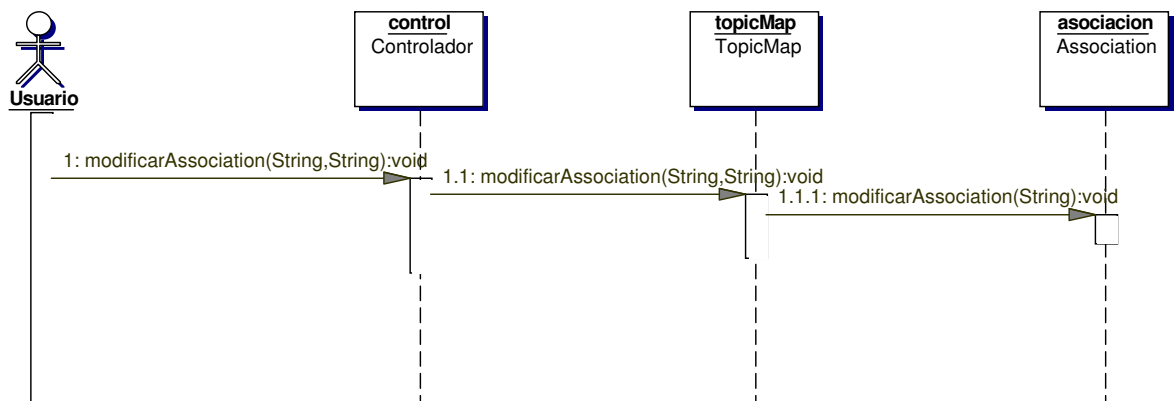


CU05: Gestión Asociaciones

CU05-3: Modificar Asociación

Caso de Uso: Modificar Asociación
Objetivo: Cambiar los datos almacenados por la asociación
Actores: Usuario(U)
Precondiciones: La asociación debe existir
Postcondiciones: Se habrán actualizado los datos que almacena la asociación
Escenario básico:
1.U: Introduce los nuevos datos de la asociación
2.S: Si los datos son correctos, actualiza los datos de la asociación; en caso contrario, muestra un mensaje de error

Diagrama de secuencia del caso de uso





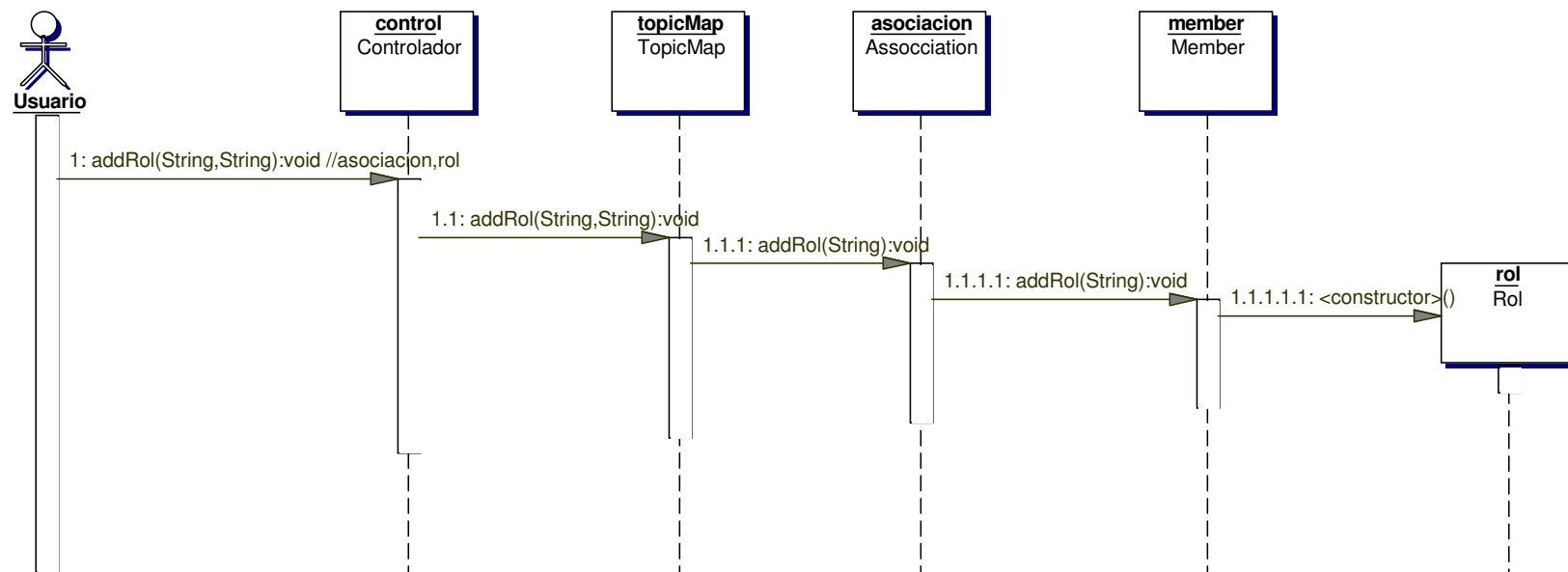
CU06: Gestión Roles

CU06-1: Crear Rol

Caso de Uso:	Crear Rol
Objetivo:	Crear una nueva rol en una asociación seleccionada
Actores:	Usuario(U)
Precondiciones:	Debe existir la asociación y el member
Postcondiciones:	Se habrá creado un nuevo rol
Escenario básico:	1.U: el usuario selecciona la asociación y el member, e introduce los datos del rol 2.S: crea el nuevo rol



Diagrama de secuencia del caso de uso



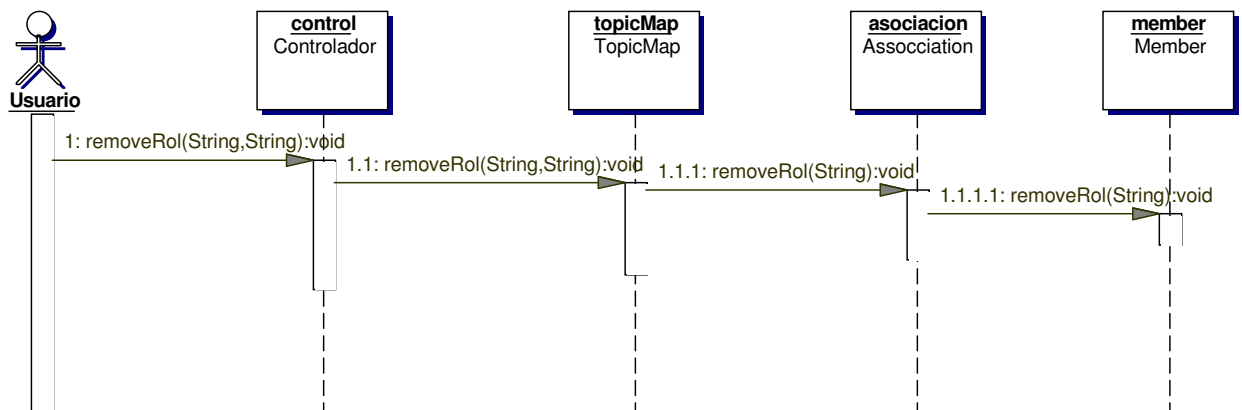


CU06: Gestión Roles

CU06-2: Eliminar Rol

Caso de Uso: Eliminar Rol
Objetivo: Eliminar el rol indicado de la asociación seleccionada
Actores: Usuario(U)
Precondiciones: La asociación, el member y el rol deben existir
Postcondiciones: Se habrá borrado el rol de la asociación seleccionada
Escenario básico:
1.U: selecciona el rol que desea borrar de la asociación seleccionada a priori
2.S: elimina el rol de la asociación

Diagrama de secuencia del caso de uso





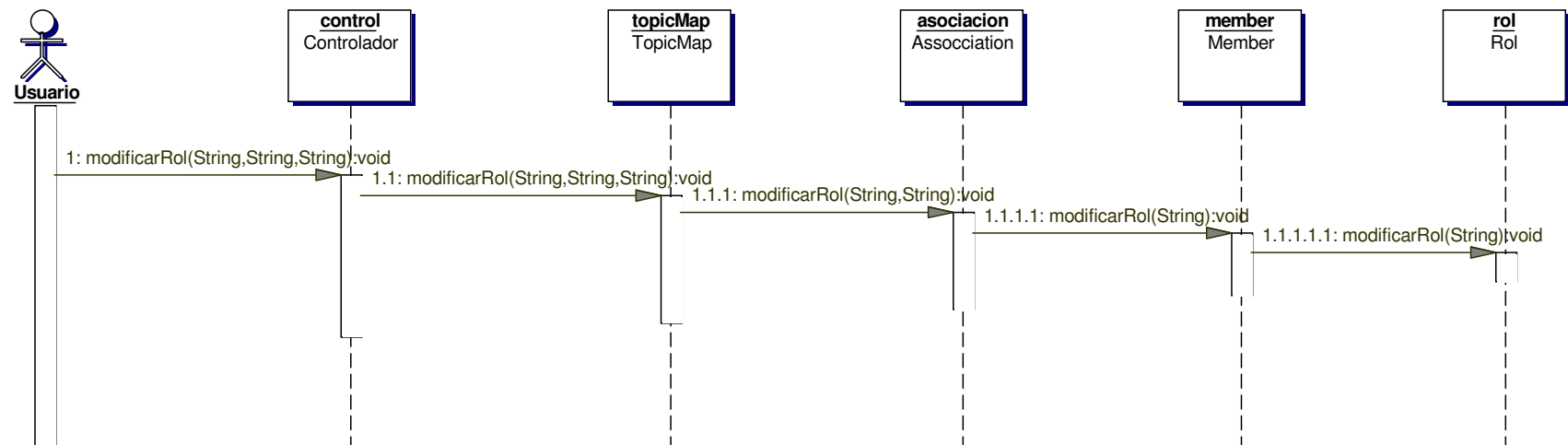
CU06: Gestión Roles

CU06-3: Modificar Rol

Caso de Uso:	Modificar Rol
Objetivo:	Cambiar los datos almacenados por el rol
Actores:	Usuario(U)
Precondiciones:	La asociación, el member y el rol deben existir
Postcondiciones:	Se habrán actualizado los datos que almacena el rol
Escenario básico:	1.U: Introduce los nuevos datos del rol 2.S: Si los datos son correctos, actualiza los datos del rol; en caso contrario, muestra un mensaje de error



Diagrama de secuencia del caso de uso



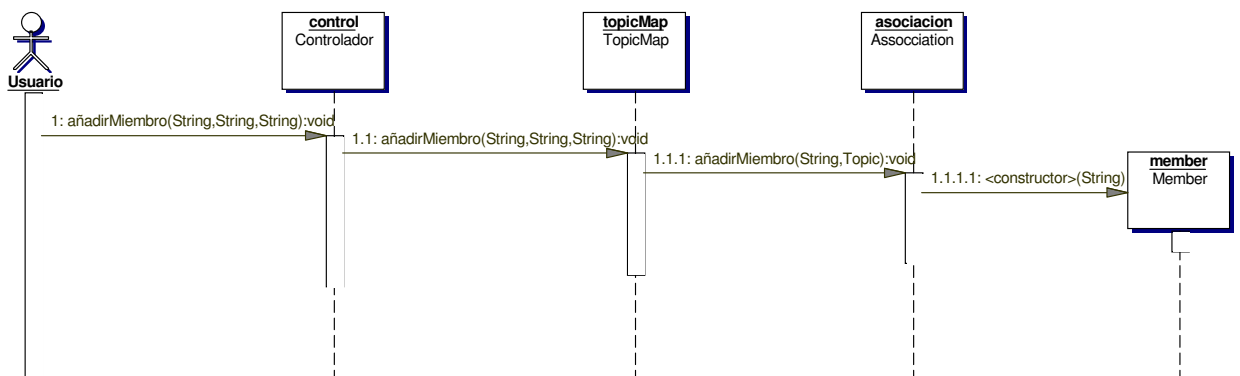


CU07: Gestión Members

CU07-1: Añadir Member

Caso de Uso: Añadir Member
Objetivo: Añadir un nuevo member a la asociación
Actores: Usuario(U)
Precondiciones: Debe existir la asociación
Postcondiciones: Se creará el member
Escenario básico:
1.U: El usuario indica los datos del nuevo member
2.S: Añade el member a la asociación

Diagrama de secuencia del caso de uso

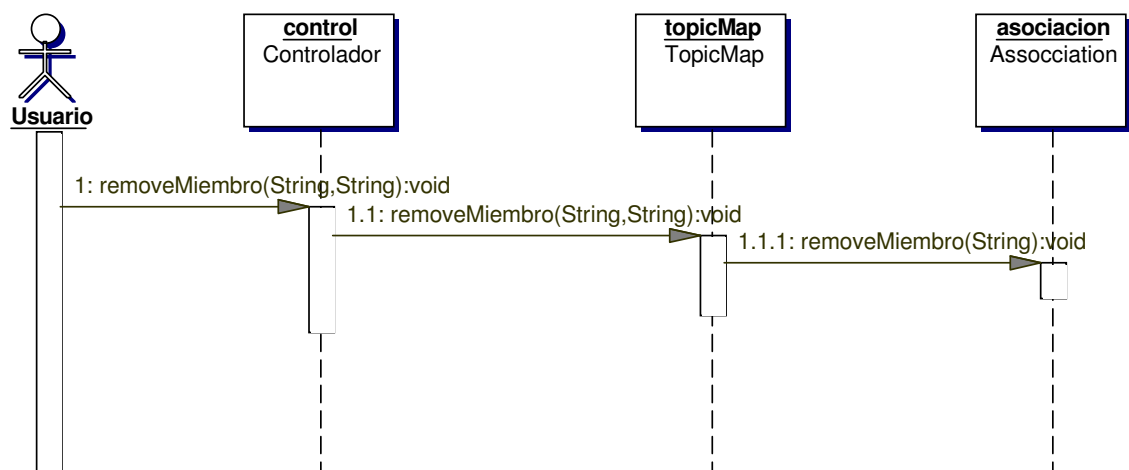


CU07: Gestión Members

CU07-2: Eliminar Member

Caso de Uso:	Eliminar Member
Objetivo:	Eliminar un member específico de una asociación
Actores:	Usuario(U)
Precondiciones:	La asociación y el member deben existir
Postcondiciones:	El member será borrado de la asociación
Escenario básico:	1.U: selecciona el member que desea borrar 2.S: elimina el member de la asociación

Diagrama de secuencia del caso de uso

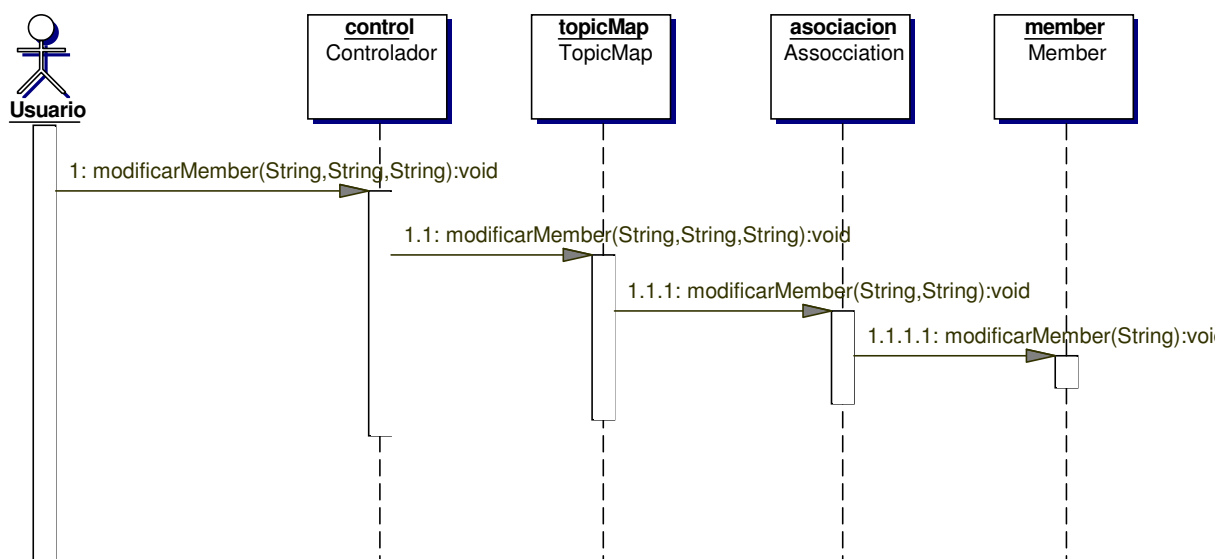


CU07: Gestión Members

CU07-3: Modificar Member

Caso de Uso: Modificar Member
Objetivo: Modificar los datos de un member
Actores: Usuario(U)
Precondiciones: La asociación y el member deben existir
Postcondiciones: El member será modificado
Escenario básico:
1.U: selecciona el member de la asociación que desea modificar
2.S: modifica los datos del member

Diagrama de secuencia del caso de uso

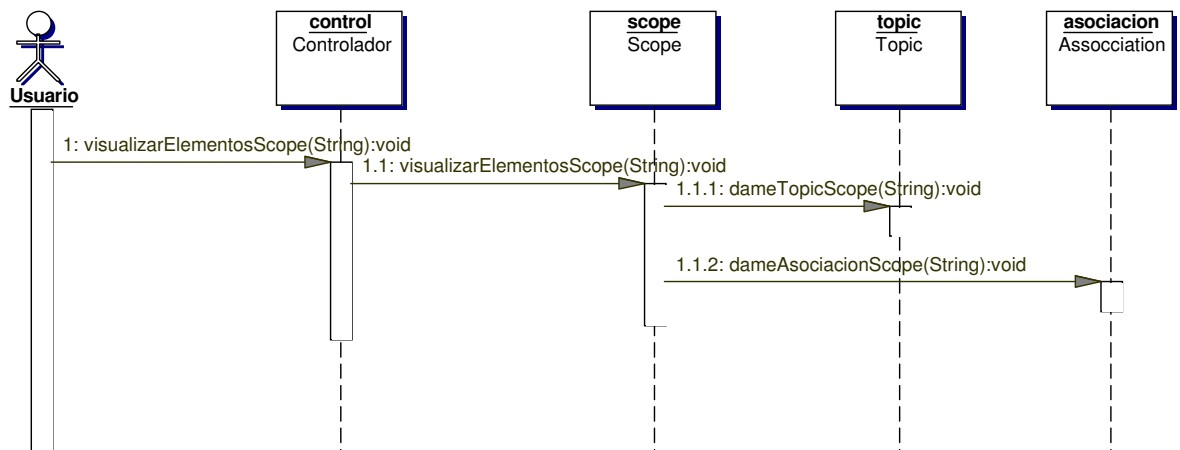


CU08: Gestión Scopes

CU08-1: Visualizar elementos Scope

Caso de Uso:	Visualizar elementos Scope
Objetivo:	Obtener una lista de topic y asociaciones que pertenecen al scope.
Actores:	Usuario(U)
Precondiciones:	-
Postcondiciones:	Se mostrarán los topics y las asociaciones del scope
Escenario básico:	1.U: selecciona el scope 2.S: muestra los datos del scope

Diagrama de secuencia del caso de uso

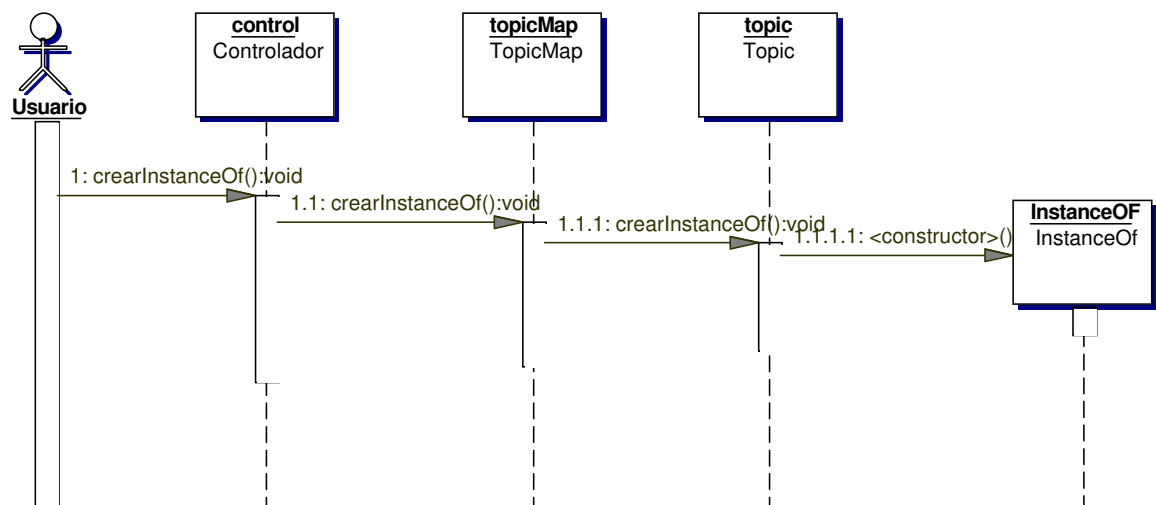


CU09: Gestión Instance Of

CU09-1: Añadir Instance Of

Caso de Uso:	Añadir Instance Of
Objetivo:	Crear un elemento instante Of
Actores:	Usuario(U)
Precondiciones:	El Topic debe existir
Postcondiciones:	El elemento instance Of se creará
Escenario básico:	1.U: indica los datos del elemento instance Of 2.S: se creará el elemento instance Of

Diagrama de secuencia del caso de uso

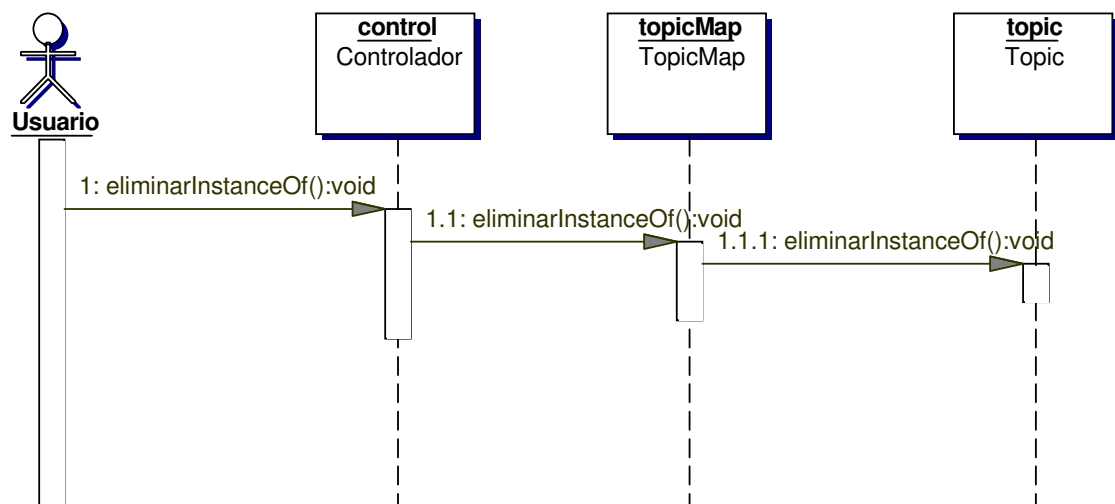


CU09: Gestión Instance Of

CU09-2: Eliminar Instance Of

Caso de Uso:	Eliminar Instance Of
Objetivo:	Eliminar un elemento instance Of del Topic seleccionado
Actores:	Usuario(U)
Precondiciones:	El Topic y el elemento instance Of deben existir
Postcondiciones:	El Topic deja de pertenecer al Scope seleccionado
Escenario básico:	1.U: indica el topic y la instancia que desea eliminar 2.S: elimina la instancia del topic

Diagrama de secuencia del caso de uso

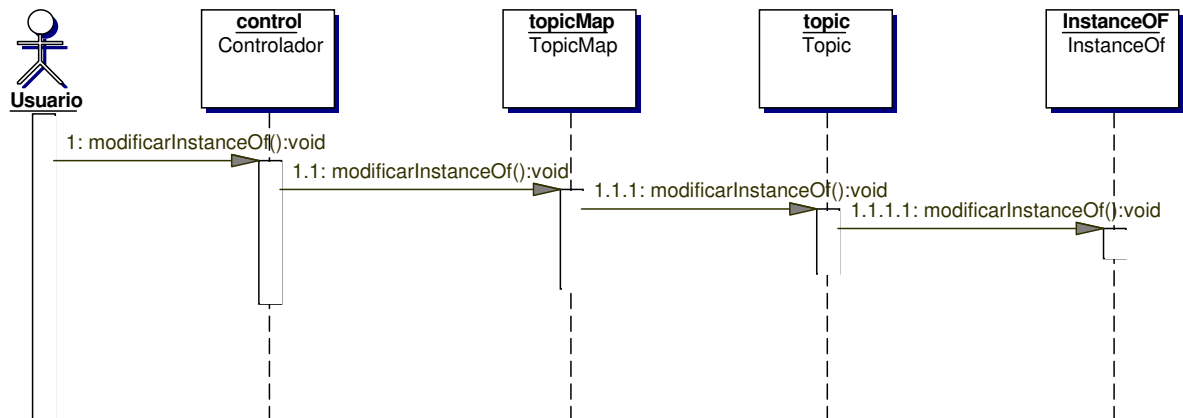




CU09: Gestión Instance Of CU09-2: Modificar Instance Of

Caso de Uso: Modificar Instance Of
Objetivo: Modificar los datos del elemento instance Of
Actores: Usuario(U)
Precondiciones: El topic y la instancia deben existir
Postcondiciones: Los datos de la instancia serán modificados
Escenario básico:
1.U: selecciona la instancia a modificar e introduce los nuevos datos
2.S: modifican los datos de la instancia

Diagrama de secuencia del caso de uso

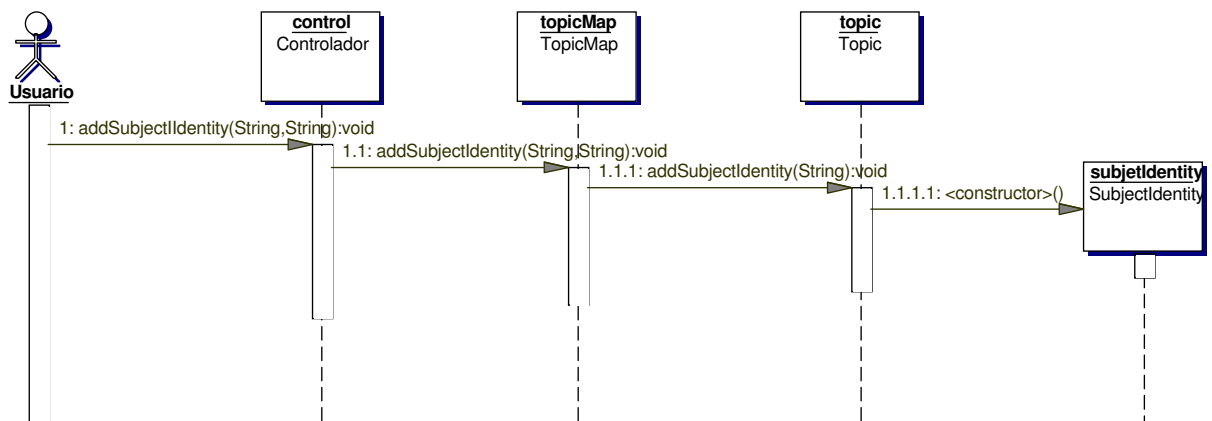


CU10: Gestión Subject Identity

CU10-1: Añadir Subject Identity

Caso de Uso: Añadir Subject Identity
Objetivo: Crear un elemento Subject Identity y asignarlo a un Topic
Actores: Usuario(U)
Precondiciones: El Topic debe existir
Postcondiciones: Se creará un Subject Identity y será asignado al Topic seleccionado
Escenario básico: 1.U: indica la información del Subject Identity que desea crear
2.S: crea y asigna el nuevo Subject Identity al Topic seleccionado

Diagrama de secuencia del caso de uso

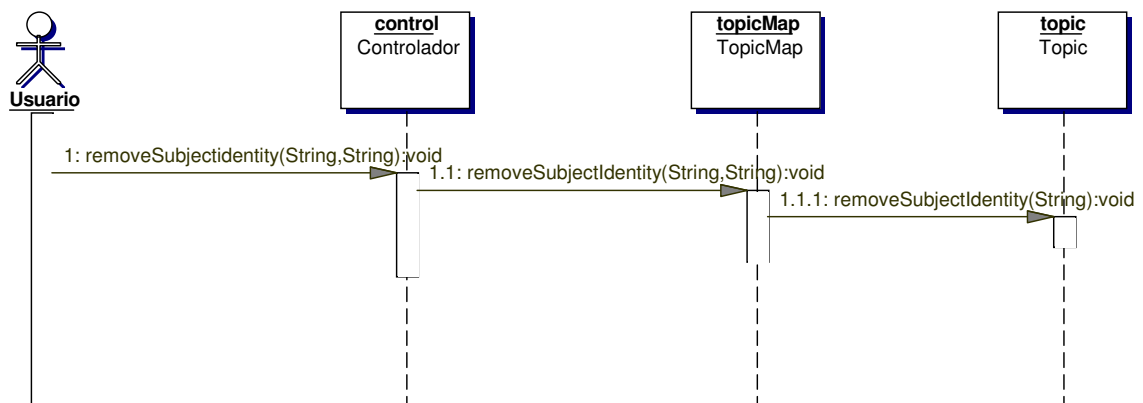


CU10: Gestión Subject Identity

CU10-2: Eliminar Subject Identity

Caso de Uso: Eliminar Subject Identity
Objetivo: Eliminar un elemento Subject Identity que está asignado a un Topic
Actores: Usuario(U)
Precondiciones: El Topic y el Subject Identity deben existir
Postcondiciones: El Subject Identity será borrado
Escenario básico: 1.U: indica el elemento Subject Identity que desea borrar
2.S: elimina el Subject Identity del Topic

Diagrama de secuencia del caso de uso

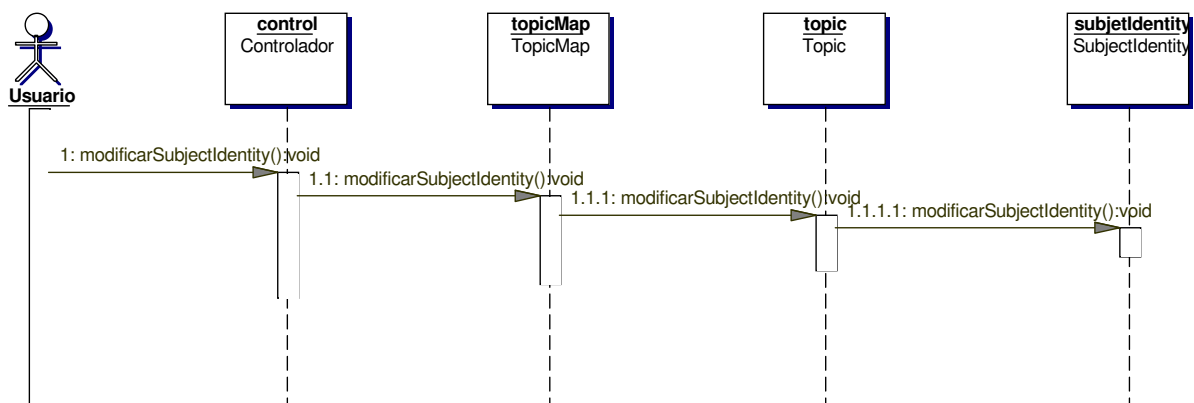


CU10: Gestión Subject Identity

CU10-3: Modificar Subject Identity

Caso de Uso: Modificar Subject Identity
Objetivo: Modificar los datos del Subject Identity seleccionado
Actores: Usuario(U)
Precondiciones: El Topic y el Subject Identity deben existir
Postcondiciones: Se modificarán los datos del Subject Identity
Escenario básico: 1.U: indica la información del Subject Identity que desea modificar
2.S: se modificarán los datos del Subject Identity

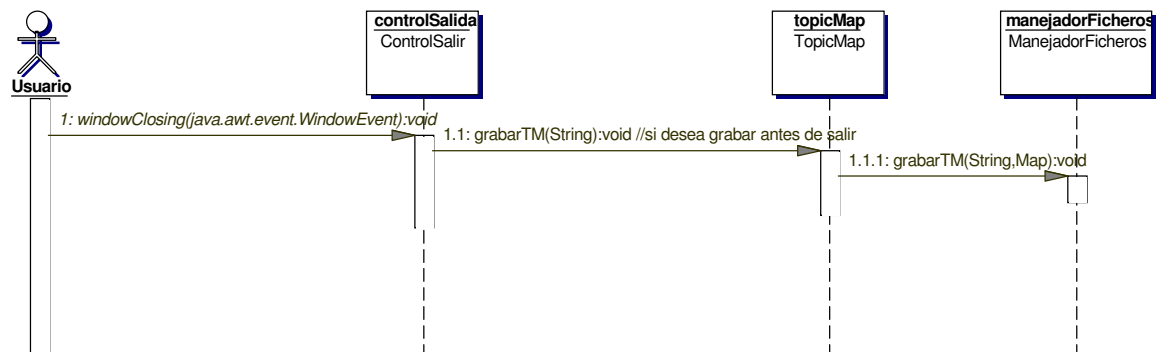
Diagrama de secuencia del caso de uso



CU11: Salir

Caso de Uso:	Salir
Objetivo:	Salir de la aplicación
Actores:	Usuario(U)
Precondiciones:	-
Postcondiciones:	El sistema se habrá cerrado
Escenario básico:	<ol style="list-style-type: none">1.U: selecciona la opción "salir del sistema"2.S: muestra al usuario una ventana que permitirá al usuario guardar antes de salir2.1S: si selecciona guardar, se producirá un guardado del TM en formato xtm3.S: finaliza el sistema

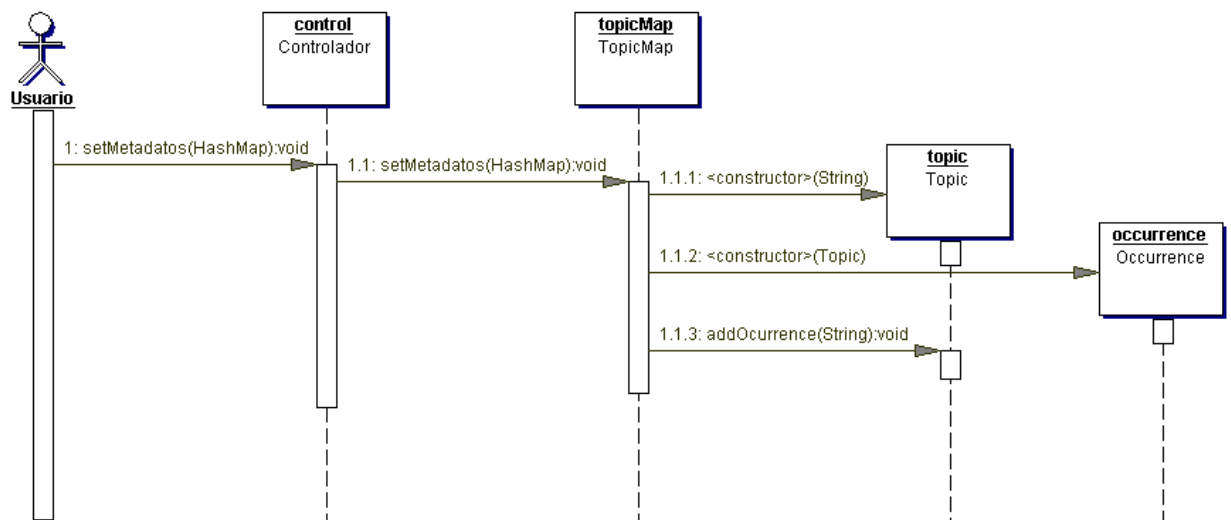
Diagrama de secuencia del caso de uso



CU12: Gestión Metadatos

Caso de Uso:	Gestión Metadatos
Objetivo:	Gestionar los metadatos del Topic Map
Actores:	Usuario(U)
Precondiciones:	-
Postcondiciones:	Se habrán guardado los metadatos del TM en forma de Topic.
Escenario básico:	<ol style="list-style-type: none">1.U: selecciona la pestaña "Metadata", inserta los metadatos y pulsa sobre el botón "Guardar".2.S: genera un topic por cada metadata guardado referenciando a Dublin Core mediante una occurrence. Se generará además un topic3.S: almacena los metadatos en memoria

Diagrama de secuencia del caso de uso



4.3. Requisitos software

El catálogo de requisitos de usuario especificará todos y cada uno de los requisitos software que deberá cumplir el sistema.

Se utilizará una tabla como la siguiente para la catalogación de requisitos:

RSXX-NN	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	Y.Y
DESCRIPCIÓN:	Descripción del requisito.

Tabla 3. Modelo de especificación de un requisito software

En dicha tabla se describirán los siguientes atributos:

- Identificación: cada requisito incluirá una identificación que será única para cada uno de ellos. Se nombran con las siglas RSXX (requisito software de XX) seguidos de un guión y dos dígitos enumerados de manera secuencial. La XX podrá ser:
 - FU: requisitos funcionales
 - DE: requisitos de desempeño (rendimiento)
 - IN: requisitos de interfaz
 - OP: requisitos operacionales
 - RE: requisitos de recursos
 - VE: requisitos de verificación
 - PA: requisitos de prueba de aceptación
 - DO: requisitos de documentación



- SE: requisitos de seguridad¹
 - PO: requisitos de portabilidad
 - CA: requisitos de calidad
 - FI: requisitos de fiabilidad
 - MA: requisitos de mantenimiento
 - SA: requisitos de seguridad²
- **Prioridad:** hace referencia al orden temporal en el que debe realizarse ese requisito. Los valores que puede tomar este campo son: prioridad alta, media y baja.
 - **Estabilidad:** señala si un requisito es o no estable. Un requisito no estable es aquel que está sujeto a posibles modificaciones, ya sea por el propio cliente o bien por otros factores que puedan afectar al sistema. Los valores de este atributo son sí o no.
 - **Claridad:** muestra la facilidad o dificultad para aplicar el requisito al sistema. Normalmente está relacionada con la dificultad para comprender el significado del requisito.
 - **Verificabilidad:** es un indicador de la dificultad o facilidad para comprobar si el requisito se cumple en el sistema final.
 - **Necesidad:** indica si un requisito es esencial o no. Se utilizará una escala de tres niveles: esencial, deseable y opcional.
 - **Versión:** permite visualizar la traza de los distintos estados por los que haya pasado un requisito para identificarlo unívocamente.

4.3.1. Requisitos funcionales

Son los requisitos que indican las funciones que el sistema podrá tener.

RSFU-01	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA

¹ Por “SE” se identificarán los requisitos de seguridad en el sentido de la palabra inglesa “security”; es decir, seguridad referida al sistema en sí.

² Por “SA” se identificarán los requisitos de seguridad en el sentido de la palabra inglesa “safety”; es decir, seguridad referida a los usuarios del sistema.



Proyecto Fin de Carrera-TMCreator

RSFU-01	
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá crear un nuevo Topic Map.

RSFU -02	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá grabar el Topic Map en un fichero con formato xtm.

RSFU -03	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá cargar un Topic Map creado anteriormente en formato xtm.



Proyecto Fin de Carrera-TMCreator

RSFU -04	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá crear nuevos Topics.

RSFU -05	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá eliminar los Topics que considere oportunos.

RSFU -06	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0



RSFU -06	
DESCRIPCIÓN:	El usuario podrá modificar la información que almacena cada Topic.

RSFU -07	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá añadir un nombre al Topic creado.

RSFU -08	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá añadir una serie de elementos instance Of al Topic creado.

RSFU -09	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA



RSFU -09	
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá eliminar los elementos instance Of del Topic que seleccione.

RSFU -10	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá modificar los datos del elemento instance Of del Topic que seleccione.

RSFU -11	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá añadir un Subject identity al Topic seleccionado.

RSFU-12



Proyecto Fin de Carrera-TMCreator

RSFU-12	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá no indicar un Subject Identity en el Topic creado

RSFU-13	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá modificar el Subject Identity del Topic seleccionado.

RSFU-14	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0



Proyecto Fin de Carrera-TMCreator

RSFU-14	
DESCRIPCIÓN:	Cuando modifique el Subject Identity, el usuario podrá añadir todos los TopicRef o SubjectIndicatorRef que desee. Además podrá añadir un resourceRef que no será obligatorio.

RSFU -15	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> No
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá indicar una o varias occurrences del Topic seleccionado.

RSFU -16	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> No
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá eliminar la occurrence seleccionada del Topic.

RSFU -17	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> No
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA



RSFU -17	
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá modificar el contenido de la Occurrence seleccionada.

RSFU -18	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	Al crear o modificar una Occurrence del Topic, el usuario podrá crear un Scope en el que encajar la Occurrence.

RSFU -19	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá indicar de 0 a n Base Names del Topic seleccionado.

RSFU -20



Proyecto Fin de Carrera-TMCreator

RSFU -20	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá eliminar el Base Name que seleccione.

RSFU -21	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá modificar los datos del Base Name.

RSFU -22	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0



Proyecto Fin de Carrera-TMCreator

RSFU -22	
DESCRIPCIÓN:	El usuario podrá indicar, al crear o modificar un Base Name, a qué scope pertenece el Base Name.

RSFU -23	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá crear asociaciones entre topics.

RSFU -24	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá eliminar las asociaciones que considere oportunas.

RSFU -25	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA



RSFU -25	
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá modificar la información que almacena cada asociación.

RSFU -26	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá indicar, al crear o modificar una asociación, a qué scope será asociada.

RSFU -27	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá crear miembros de la asociación.

RSFU -28



Proyecto Fin de Carrera-TMCreator

RSFU -28	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá eliminar miembros de la asociación seleccionada.

RSFU -29	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá modificar los datos de cada uno de los miembros de la asociación.

RSFU -30	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0



RSFU -30	
DESCRIPCIÓN:	El usuario podrá visualizar los topics que estén ligados al scope seleccionado.

RSFU -31	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> No
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá visualizar las asociaciones que estén ligadas al scope seleccionado.

RSFU -32	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> No
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
VERSIÓN	1.0
DESCRIPCIÓN:	El usuario podrá añadir metadatos al Topic Map.

4.3.2. Requisitos de rendimiento

4.3.3. Requisitos de interfaz



Proyecto Fin de Carrera-TMCreator

RSIN-01	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
FUENTE:	Cliente
VERSIÓN	1.0
DESCRIPCIÓN:	El sistema recibirá flujos de datos procedentes del teclado y ratón.

RSIN-02	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
FUENTE:	Cliente
VERSIÓN	1.0
DESCRIPCIÓN:	La interfaz gráfica de usuario enviará los flujos de datos al monitor del entorno de funcionamiento a través de su propio sistema.

RSIN-03	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
FUENTE:	Cliente



VERSIÓN	1.0
DESCRIPCIÓN:	El componente que permite almacenar los topic maps en formato xtm leerá y almacenará ficheros en el disco del entorno de ejecución.

RSIN-04	
PRIORIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> No
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
FUENTE:	Cliente
VERSIÓN	1.0
DESCRIPCIÓN:	Todos los componentes internos del sistema se comunicarán con la máquina virtual de Java durante la ejecución.

4.3.4. Requisitos operacionales

4.3.5. Requisitos de recursos

4.3.6. Requisitos de verificación

4.3.7. Requisitos de pruebas y aceptación

4.3.8. Requisitos de documentación

RSDO-01	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> No
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
FUENTE:	Cliente



VERSIÓN	1.0
DESCRIPCIÓN:	El código generado debe estar totalmente documentado para su fácil entendimiento, en caso de una futura modificación.

4.3.9. Requisitos de seguridad

RSSE-01	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
FUENTE:	Cliente
VERSIÓN	1.0
DESCRIPCIÓN:	No será necesaria ninguna autenticación de usuario para el uso del sistema.

4.3.10. Requisitos de portabilidad

RSPO-01	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
FUENTE:	Cliente
VERSIÓN	1.0



DESCRIPCIÓN:	El sistema informático se desarrollará con el lenguaje de programación Java.
--------------	--

RSPO-02	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
FUENTE:	Cliente
VERSIÓN	1.0
DESCRIPCIÓN:	El sistema informático deberá funcionar bajo el sistema operativo Microsoft Windows 2k/XP

4.3.11. Requisitos de calidad

4.3.12. Requisitos de fiabilidad

4.3.13. Requisitos de mantenimiento

4.3.14. Requisitos de seguridad (safety)

RSSA-01	
PRIORIDAD: <input type="checkbox"/> ALTA <input checked="" type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	ESTABILIDAD: <input checked="" type="checkbox"/> SÍ <input type="checkbox"/> NO
CLARIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	VERIFICABILIDAD: <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
NECESIDAD: <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL	
FUENTE:	Cliente
VERSIÓN	1.0



DESCRIPCIÓN:	El sistema deberá presentar una resolución ajustada para evitar problemas visuales al usuario, tales como vista cansada, o el simple hecho de forzar la vista ante la sobrecarga de información en la ventana con elementos demasiado pequeños o irritantes (visualmente hablando)
--------------	--

4.4. Diagrama de clases inicial

A continuación se expondrá el diagrama de paquetes resultante del análisis, y posteriormente, se presentará el diagrama de clases de cada paquete. Hay que recordar que estos diagramas son una primera aproximación a los diagramas finales.

4.4.1. Diagrama de paquetes

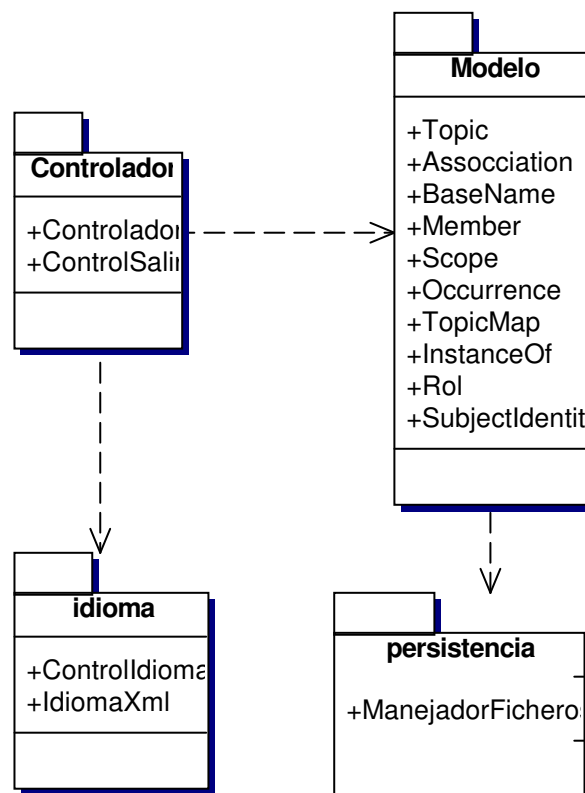


Figura 1. Diagrama de paquetes inicial

4.4.2. Diagrama de clases del paquete controlador

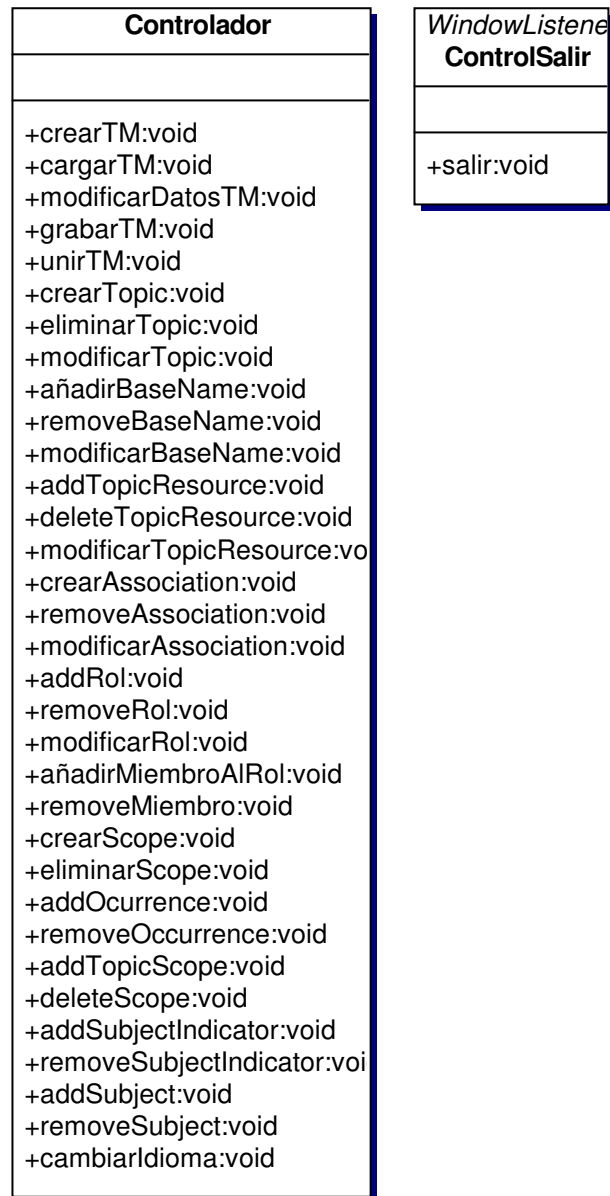


Figura 2. Diagrama de clases inicial del paquete controlador

4.4.3. Diagrama de clases del paquete modelo

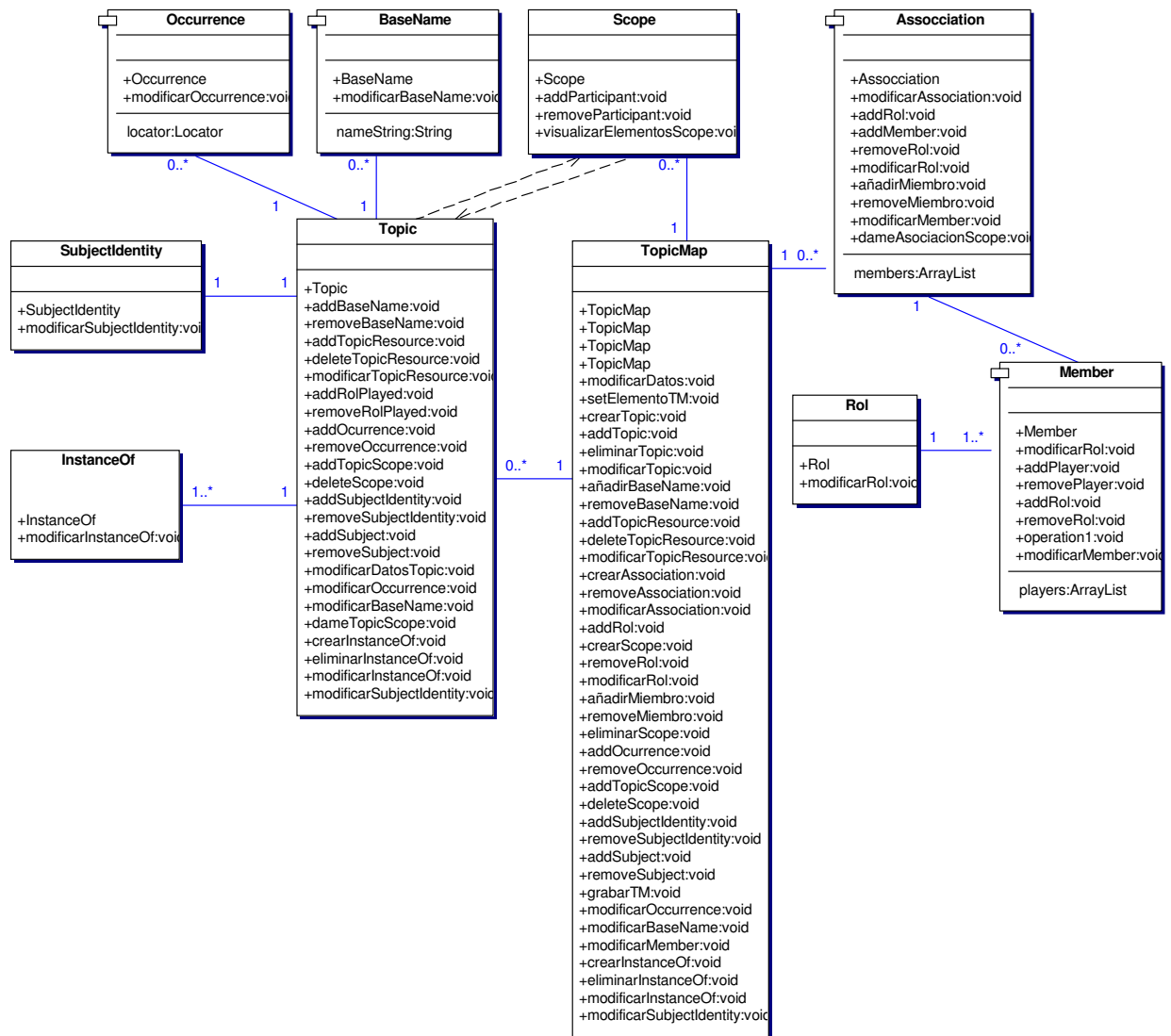


Figura 3. Diagrama de clases inicial del paquete modelo



4.4.4. Diagrama de clases del paquete persistencia

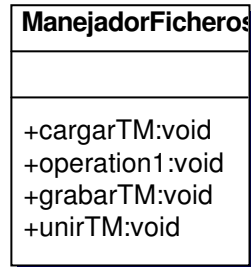


Figura 4. Diagrama de clases inicial del paquete persistencia

5. Diseño del sistema

5.1. Diseño arquitectónico

El contexto y diseño bajo el que nos situamos para el sistema, se ubica en el modelado orientado a objetos.

La arquitectura elegida para el diseño de la aplicación es el Model-View-Controller (a partir de ahora MVC). Consideramos oportuno señalar que es la más adecuada, ya que concuerda bastante con el modelo orientado a objetos. Como aspectos negativos, tiene un inicio más lento, al tener que pensar y elaborar una correcta separación de las partes principales del patrón.

Este modelo de arquitectura presenta varias ventajas:

- Hay una clara separación entre los componentes de un programa, lo cual permite implementarlo por separado.
- Hay un API muy bien definido; cualquiera que use el API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.

El esquema general del MVC de la mano de Sun Microsystems es el siguiente:

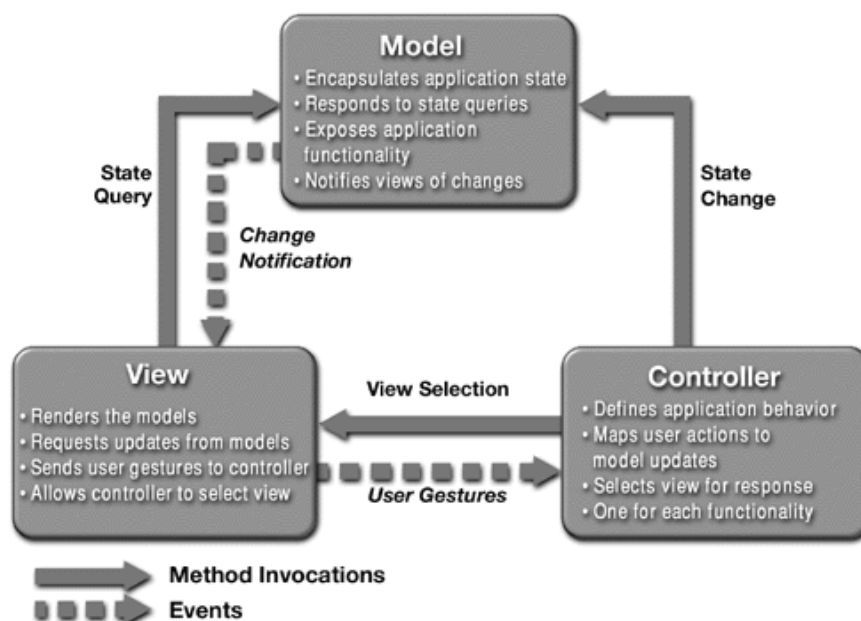


Figura 5. MVC de Sun Microsystems

Al incorporar el modelo de arquitectura MVC a un diseño, las piezas de un programa se pueden construir por separado y, posteriormente, pueden unirse en tiempo de ejecución. Si se observa con posterioridad que uno de los componentes funciona mal, puede reemplazarse sin que las otras piezas se vean afectadas.

Conviene aclarar que la arquitectura elegida se modificará levemente para añadir un módulo adicional: el subsistema de persistencia. Esta modificación se realiza para desacoplarlo aún más, y poder sustituir fácilmente el sistema de persistencia sin tener que modificar el modelo. El resultado es el siguiente:

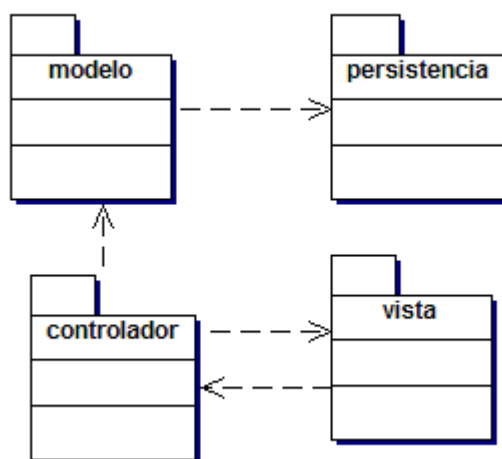


Figura 6. Arquitectura MVC

5.1.1. Descripción de la descomposición

El objetivo de esta sección es definir las divisiones y relaciones entre los distintos subsistemas que conformarán el sistema final. Para llevar a cabo dicha tarea, se recurrirá tanto a diagramas UML como a una breve descripción de cada uno de los subsistemas.

La forma de identificar a los niveles será un número desde 1 hasta el más bajo. Si hubiera varios subsistemas en el mismo nivel, se usará X.Y, siendo X el nivel de los hijos del subsistema padre e Y un número identificativo para referirse a cada hijo.

Nivel 1: Arquitectura del sistema.

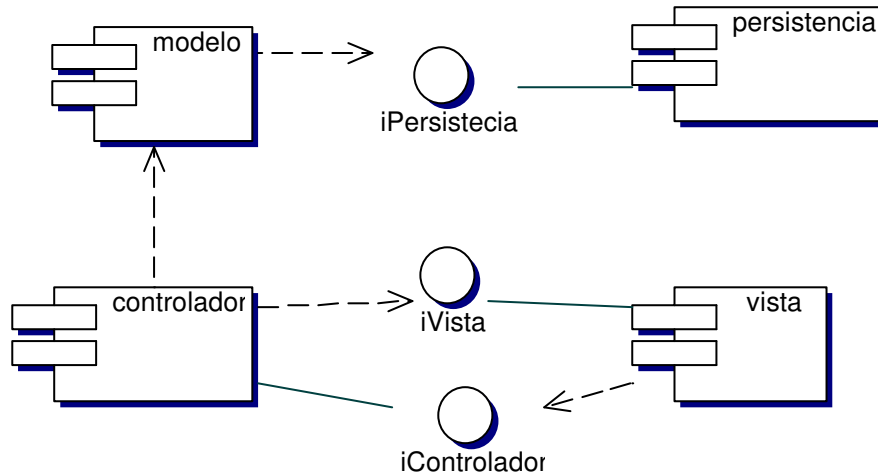


Figura 7. Arquitectura del sistema

El componente *vista* incluiría directamente las clases encargadas de todas las interfaces gráficas de usuario de la aplicación, sin poder ser dividido en más subsistemas. El *controlador*, del mismo modo, incluirá una clase que llamará a cada uno de los subsistemas del modelo, sin ser dividido posteriormente. El *modelo*, al contrario, presentará una división más precisa en subsistemas y componentes, que será indicada en el siguiente nivel.

La *persistencia*, al igual que el controlador y vista, no se dividirá en nuevos subsistemas.

Nivel 2: Modelo.

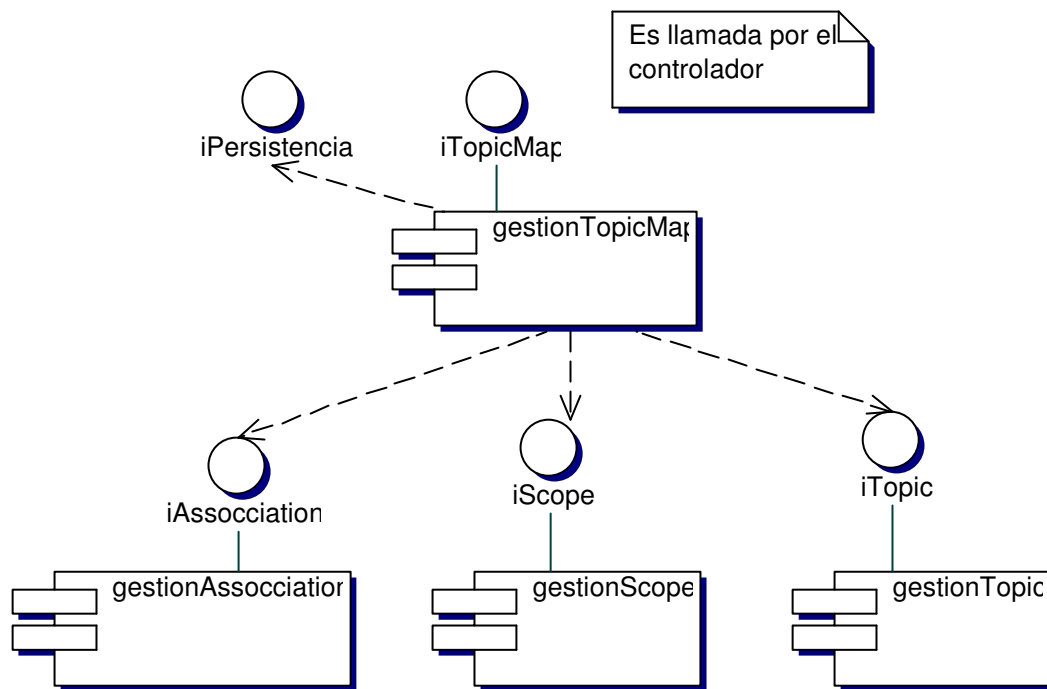


Figura 8. Arquitectura del modelo

Los elementos del modelo son: *gestionTopicMap*, *gestionAssociation*, *gestionScope* y *gestionTopic*. El elemento *gestionTopicMap* tendrá dependencias con el subsistema de persistencia en su interfaz de comunicación (dicha interfaz se adjunta en este diagrama como objeto remoto para su constatación).

El componente *gestionTopicMap* es el componente que lleva el control sobre todo el modelo; es el encargado de relacionar todo el modelo y es el único que ofrece una interfaz de comunicación con otros componentes externos.

Los otros tres componentes se encargarán de gestionar los distintos elementos del estándar de Topic Map. El componente *gestionAssociation* se encargará de gestionar las asociaciones; el componente *gestionScope* se encargará de gestionar los Scope del sistema; y el componente *gestionTopic*, de gestionar los Topic.



5.1.2. Descripción de componentes

5.1.2.1 Modelo.gestionTopicMap

Tipo

El componente pertenece al subsistema modelo, por lo tanto su lógica queda definida con la inclusión del mismo en su correspondiente módulo. Las características físicas definidas en terminología de implementación, ubican este componente como una tarea que podría realizar el usuario.

Propósito

El componente gestionTopicMap implementa los siguientes requisitos: RSFU-01, RSFU -02, RSFU -03, RSFU -04, RSFU -05, RSFU -06, RSFU -07, RSFU -08, RSFU -09, RSFU -10, RSFU-11, RSFU -12, RSFU -13, RSFU -14, RSFU -15, RSFU -16, RSFU -17, RSFU -18, RSFU -19, RSFU -20, RSFU -21, RSFU -22, RSFU -23, RSFU -24, RSFU -25, RSFU -26, RSFU -27, RSFU -28, RSFU -29, RSFU -30, RSFU -31, RSFU -32.

Función

El componente se encarga de implementar y ofrecer una interfaz para la gestión de todas las operaciones que se pueden realizar con un Topic Map (crear Topic, crear asociaciones, etc.).

Subordinados

Contiene un conjunto de clases usadas para realizar las distintas gestiones y operaciones descritas en la función del componente. Este conjunto de clases será estudiado en la fase posterior de diseño.

Dependencias

El componente de gestión analizado depende del componente de persistencia de datos, que le permite cargar y grabar Topic Maps en formato xtm. Además, depende de los componentes gestionAssociation, gestionTopic y gestionScope para poder realizar su funcionalidad.

Interfaces

El componente “ofrece” una interfaz de comunicación hacia el exterior (InterfazGestionTM), por medio de la cual los servicios prestados son accesibles para otros componentes del sistema. Estos servicios están estructurados según el siguiente diagrama:

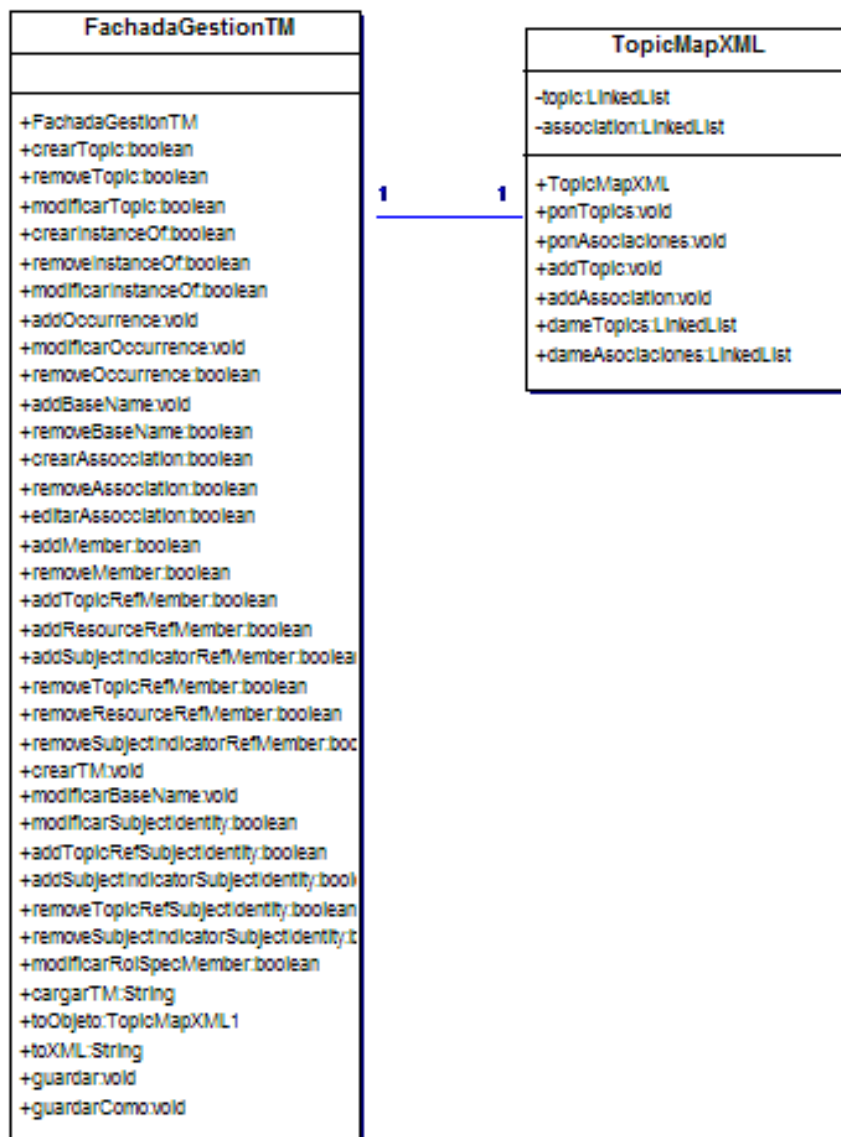


Figura 9. Componente gestionTopicMap

La FachadaGetionTM tiene las siguientes operaciones de “acceso” al componente:

- **crearTopic:** crea un nuevo Topic con las características deseadas.
- **removeTopic:** elimina el Topic seleccionado del Topic Map.
- **modificarTopic:** permite cambiar los datos informativos del Topic.
- **crearInstanceOf:** crea un nuevo elemento instanceOf a un Topic seleccionado.
- **removeInstanceOf:** elimina el instanceOf elegido del Topic seleccionado.
- **modificarInstanceOf:** modifica los datos del instanceOf elegido del Topic seleccionado.
- **addOccurrence:** añade una nueva occurrence a un Topic seleccionado.
- **modificarOccurrence:** modifica los datos de la occurrence elegida del Topic seleccionado.
- **removeOccurrence:** elimina la occurrence elegida del Topic seleccionado.
- **addBaseName:** añade un nuevo BaseName a un Topic seleccionado.
- **removeBaseName:** elimina el BaseName elegido del Topic seleccionado.
- **modificarBaseName:** modifica el BaseName elegido del Topic seleccionado.
- **crearAssociation:** crea una nueva asociación.
- **removeAssociation:** elimina la asociación seleccionada.
- **editarAssociation:** modifica los datos de la asociación.
- **addMember:** añade un nuevo miembro a la asociación.
- **removeMember:** elimina el miembro seleccionado de la asociación.
- **addTopicRefMember:** crea un TopicRef al miembro de la asociación seleccionado.
- **addResourceRefMember:** crea un ResourceRef al miembro de la asociación seleccionado.
- **addSubjectIndicatorRefMember:** crea un SubjectIndicatorRef al miembro de la asociación seleccionado.
- **removeTopicRefMember:** elimina el TopicRef elegido del miembro de la asociación.
- **removeResourceRefMember:** elimina el ResourceRef elegido del miembro de la asociación.
- **removeSubjectIndicatorRefMember:** elimina el SubjectIndicatorRef elegido del miembro de la asociación.
- **dameTopicsScope:** devuelve los Topics que son del Scope seleccionado.
- **dameAsociacionesScope:** devuelve las asociaciones que son del Scope seleccionado.
- **modificarSubjectIdentity:** modifica los datos del SubjectIdentity seleccionado.
- **addTopicRefSubjectIdentity:** añade un TopicRef al SubjectIdentity.
- **addSubjectIndicatorSubjectIdentity:** añade un SubjectIndicator al SubjectIdentity.
- **removeTopicRefSubjectIdentity:** elimina el TopicRef del SubjectIdentity.
- **removeSubjectIndicatorSubjectIdentity:** elimina el SubjectIndicator del SubjectIdentity.
- **modificarRolSpecMember:** modifica el rol del miembro de la asociación seleccionado.
- **crearTM:** método que permite al usuario crear un nuevo Topic Map.
- **cargarTM:** carga los datos de un fichero xtm.
- **toObjeto:** parsea los datos xtm y lo transforma en objetos.
- **toXML:** método que transforma los objetos en xml.



- **guardar:** guarda el xml en un fichero físico.
- **guardarComo:** guarda el xml en un fichero físico, posibilitando el cambio de nombre del fichero.

Recursos

No necesita ningún recurso extra, a excepción del propio sistema.

Referencias

Ninguna

Procesamiento

El componente recibirá las peticiones a través de la clase que implemente la interfaz dada, y transmitirá la petición al “subcomponente” de gestión correspondiente, realizando éste último las operaciones pertinentes.



5.1.2.2 Modelo.gestionTopic

Tipo

El componente pertenece al subsistema modelo, por lo tanto su lógica queda definida con la inclusión del mismo en su correspondiente módulo. Las características físicas definidas en terminología de implementación ubican este componente como una tarea que podría realizar el usuario.

Propósito

El componente gestionTopic implementa los siguientes requisitos software: RSFU -04, RSFU -05, RSFU -06, RSFU -07, RSFU -08, RSFU -09, RSFU -10, RSFU-11, RSFU -12, RSFU -13, RSFU -14, RSFU -15, RSFU -16, RSFU -17, RSFU -18, RSFU -19, RSFU -20, RSFU -21, RSFU -22, RSFU -32.

Función

El componente se encarga de implementar y ofrecer una interfaz para la gestión de todas las operaciones que se pueden realizar con un Topic y todos los elementos que se le pueden asignar.

Subordinados

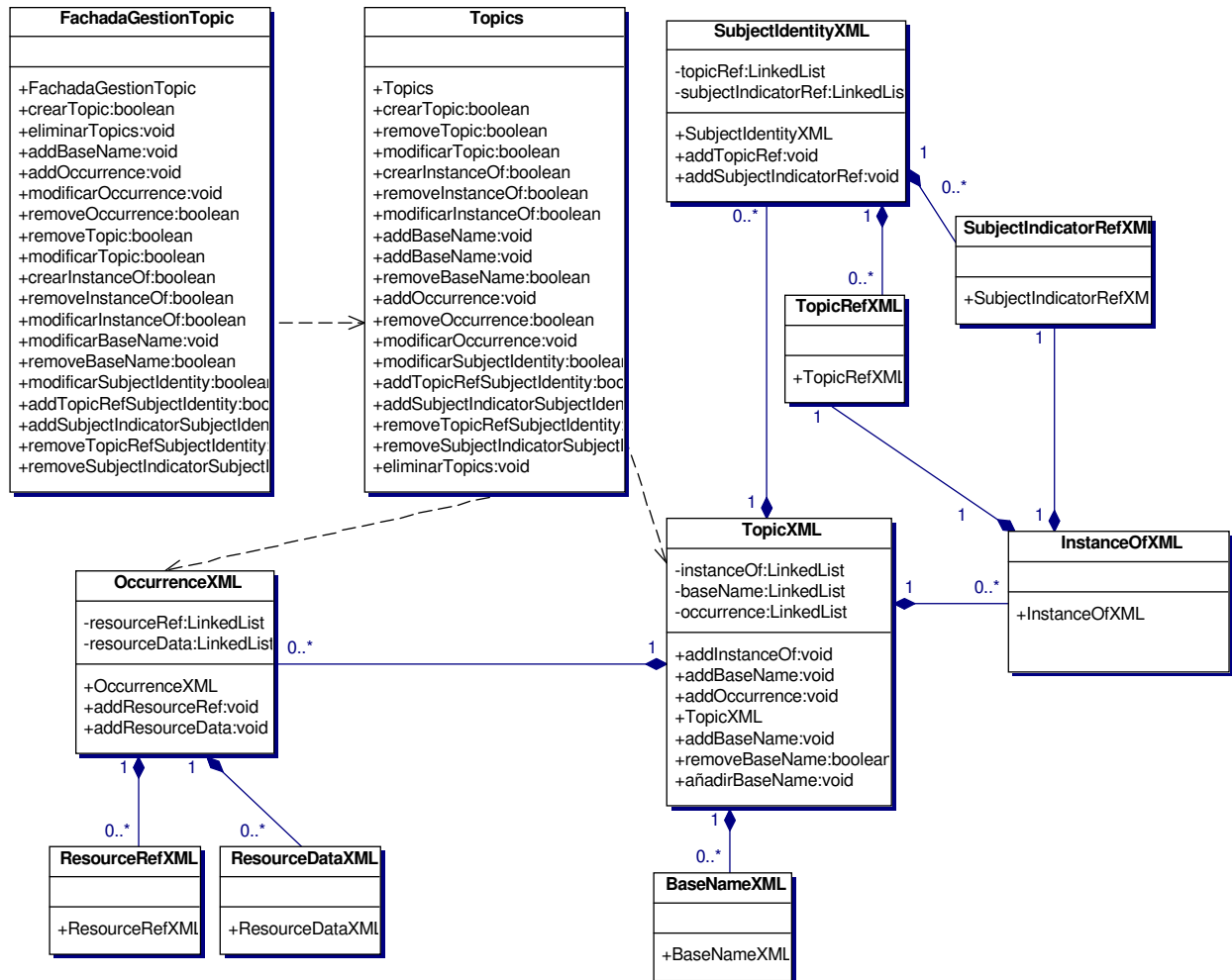
Contiene un conjunto de clases usadas para realizar las distintas gestiones y operaciones descritas en la función del componente. Este conjunto de clases será estudiado en la fase posterior de diseño.

Dependencias

El componente tiene dependencias con el componente gestionScope.

Interfaces

El componente “ofrece” una interfaz de comunicación hacia el exterior (InterfazTopic), por medio de la cual los servicios prestados son accesibles para otros componentes del sistema. Estos servicios están estructurados según el siguiente diagrama:



La FachadaGestionTopic tiene las siguientes operaciones de “acceso” al componente:

- **crearTopic:** crea un nuevo Topic con las características deseadas.
- **eliminarTopic:** elimina el Topic seleccionado del Topic Map.
- **modificarTopic:** permite cambiar los datos informativos del Topic.
- **crearInstanceOf:** crea un nuevo instanceOf a un Topic seleccionado.
- **removeInstanceOf:** elimina el instanceOf elegido del Topic seleccionado.
- **modificarInstanceOf:** modifica los datos del instanceOf elegido del Topic seleccionado.
- **addOccurrence:** añade una nueva occurrence a un Topic seleccionado.
- **modificarOccurrence:** modifica los datos de la occurrence elegida del Topic seleccionado.



- **removeOccurrence:** elimina la occurrence elegida del Topic seleccionado.
- **addBaseName:** añade un nuevo BaseName a un Topic seleccionado.
- **removeBaseName:** elimina el BaseName elegido del Topic seleccionado.
- **modificarBaseName:** modifica el BaseName elegido del Topic seleccionado.
- **modificarSubjectIdentity:** modifica los datos del SubjectIdentity seleccionado.
- **addTopicRefSubjectIdentity:** añade un TopicRef al SubjectIdentity.
- **addSubjectIndicatorSubjectIdentity:** añade un SubjectIndicator al SubjectIdentity.
- **removeTopicRefSubjectIdentity:** elimina el TopicRef del SubjectIdentity.
- **removeSubjectIndicatorSubjectIdentity:** elimina el SubjectIndicator del SubjectIdentity.

Recursos

No necesita ningún recurso extra, a excepción del propio sistema.

Referencias

Ninguna

Procesamiento

El componente recibirá las peticiones a través de la clase que implemente la interfaz dada y realizará las operaciones pertinentes, devolviendo el resultado.



5.1.2.3 Modelo.gestionAssociation

Tipo

El componente pertenece al subsistema modelo, por lo tanto su lógica queda definida con la inclusión del mismo en su correspondiente módulo. Las características físicas definidas en terminología de implementación, ubican este componente como una tarea que podría realizar el usuario.

Propósito

El componente GestiónAssociation implementa los siguientes requisitos software: RSFU-23, RSFU -24, RSFU -25, RSFU -26, RSFU -27, RSFU -28, RSFU -29.

Función

El componente se encarga de implementar y ofrecer una interfaz para la gestión de todas las operaciones que se pueden realizar con una asociación en el estándar de Topic Map.

Subordinados

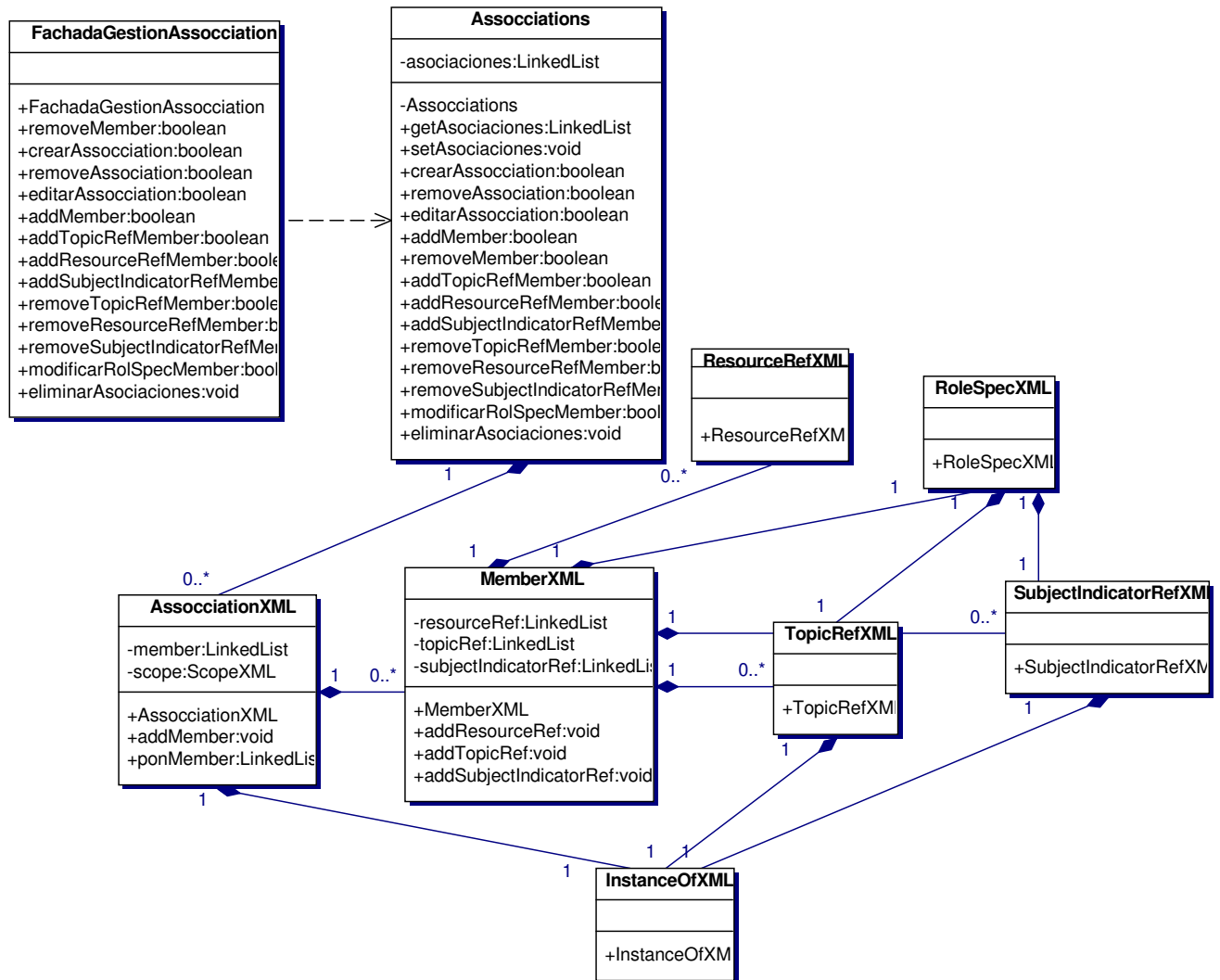
Contiene un conjunto de clases usadas para realizar las distintas gestiones y operaciones descritas en la función del componente. Este conjunto de clases será estudiado en la fase posterior de diseño.

Dependencias

Tiene dependencias con el componente gestionScope.

Interfaces

El componente “ofrece” una interfaz de comunicación hacia el exterior (InterfazAssociation), por medio de la cual los servicios prestados son accesibles para otros componentes del sistema. Estos servicios están estructurados según el siguiente diagrama:



La InterfazAssociation tiene las siguientes operaciones de “acceso” al componente:

- **crearAssociation:** crea una nueva asociación.
- **removeAssociation:** elimina la asociación seleccionada.
- **editarAssociation:** modifica los datos de la asociación.
- **addMember:** añade un nuevo miembro a la asociación.
- **removeMember:** elimina el miembro seleccionado de la asociación.
- **addTopicRefMember:** crea un TopicRef al miembro de la asociación seleccionado.
- **addResourceRefMember:** crea un ResourceRef al miembro de la asociación seleccionado.



- **addSubjectIndicatorRefMember:** crea un SubjectIndicatorRef al miembro de la asociación seleccionado.
- **removeTopicRefMember:** elimina el TopicRef elegido del miembro de la asociación.
- **removeResourceRefMember:** elimina el ResourceRef elegido del miembro de la asociación.
- **removeSubjectIndicatorRefMember:** elimina el SubjectIndicatorRef elegido del miembro de la asociación.
- **eliminarAsociaciones:** método que elimina las asociaciones almacenadas.

Recursos

No necesita ningún recurso extra, a excepción del propio sistema.

Referencias

Ninguna

Procesamiento

El componente recibirá las peticiones a través de la clase que implemente la interfaz dada y realizará las operaciones pertinentes, devolviendo el resultado.



5.1.2.4 Modelo.gestionScope

Tipo

El componente pertenece al subsistema modelo, por lo tanto su lógica queda definida con la inclusión del mismo en su correspondiente módulo. Las características físicas definidas en terminología de implementación ubican este componente como una tarea que podría realizar el usuario.

Propósito

El componente gestionScope implementa los siguientes requisitos software: RSFU -30, RSFU -31.

Función

El componente se encarga de implementar y ofrecer una interfaz para la gestión de todas las operaciones que se pueden realizar con los Scope.

Subordinados

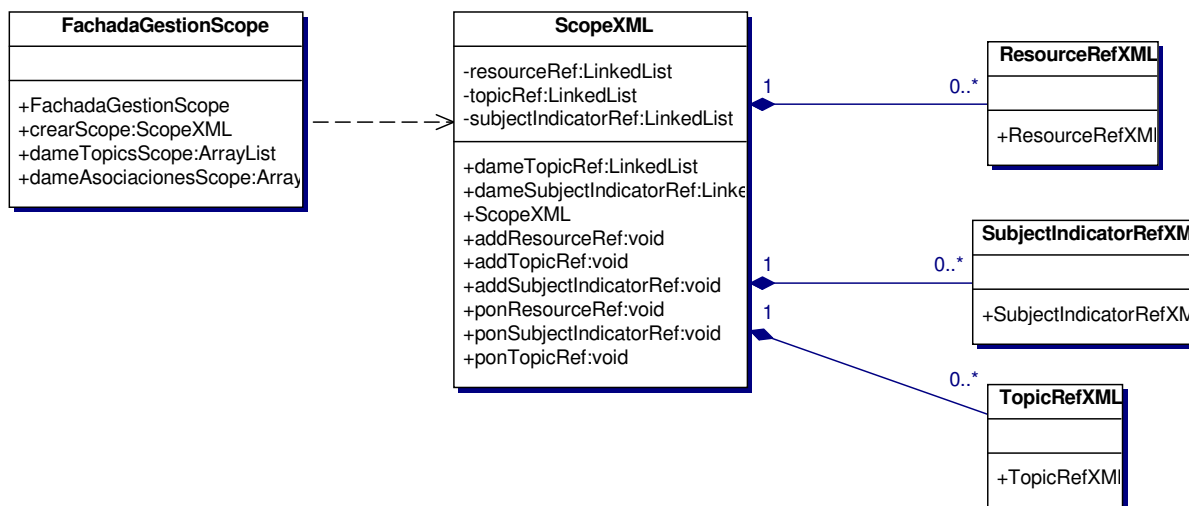
Contiene un conjunto de clases usadas para realizar las distintas gestiones y operaciones descritas en la función del componente. Este conjunto de clases será estudiado en la fase posterior de diseño.

Dependencias

No tiene dependencias con componentes externos.

Interfaces

El componente “ofrece” una interfaz de comunicación hacia el exterior (InterfazScope), por medio de la cual los servicios prestados son accesibles para otros componentes del sistema. Estos servicios están estructurados según el siguiente diagrama:



La InterfazScope tiene las siguientes operaciones de “acceso” al componente:

- **crearScope:** crea un nuevo Scope en el Topic Map.
- **dameTopicsScope:** devuelve los Topics que son del Scope seleccionado.
- **dameAsociacionesScope:** devuelve las asociaciones que son del Scope seleccionado.

Recursos

No necesita ningún recurso extra, a excepción del propio sistema.

Referencias

Ninguna

Procesamiento

El componente recibirá las peticiones a través de la clase que implemente la interfaz dada y realizará las operaciones pertinentes, devolviendo el resultado.



5.1.2.5 Persistencia

Tipo

El componente pertenece al subsistema modelo, por lo tanto su lógica queda definida con la inclusión del mismo en su correspondiente módulo. Las características físicas definidas en terminología de implementación ubican este componente como una tarea que podría realizar el usuario.

Propósito

El componente persistencia implementa los siguientes requisitos software: RSFU -02, RSFU -03.

Función

El componente se encarga de implementar y ofrecer una interfaz que permite cargar y grabar ficheros con extensión xtm.

Subordinados

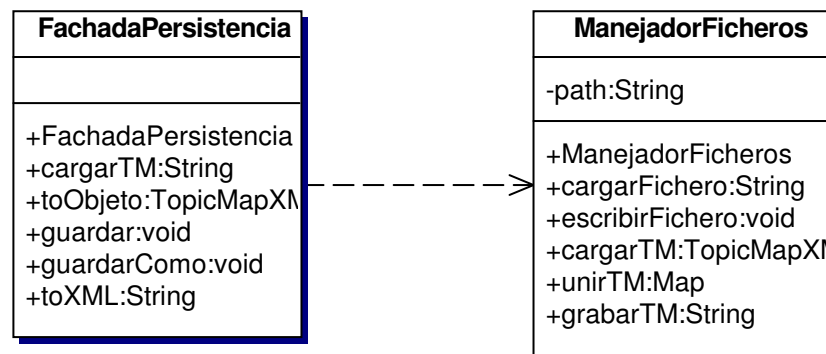
Contiene un conjunto de clases usadas para realizar las distintas gestiones y operaciones descritas en la función del componente. Este conjunto de clases será estudiado en la fase posterior de diseño.

Dependencias

El componente tiene dependencias con la librería de apache Commons Betwixt 0.8. Betwixt es una librería que permite mapear beans a XML, y al contrario. Se ha utilizado la librería Commons Betwixt para exportar el estado de los recursos a un formato XML y para recuperar los datos XML.

Interfaces

El componente “ofrece” una interfaz de comunicación hacia el exterior (InterfazPersistencia), por medio de la cual los servicios prestados son accesibles para otros componentes del sistema. Estos servicios están estructurados según el siguiente diagrama:



La InterfazPersistencia tiene las siguientes operaciones de “acceso” al componente:

- **cargarTM:** carga los datos de un fichero xtm.
- **toObjeto:** parsea los datos xtm y lo transforma en objetos.
- **toXML:** método que transforma los objetos en xml.
- **guardar:** guarda el xml en un fichero físico.
- **guardarComo:** guarda el xml en un fichero físico posibilitando el cambio de nombre del fichero.

Recursos

No necesita ningún recurso extra, a excepción del propio sistema.

Referencias

Ninguna

Procesamiento

El componente recibirá las peticiones a través de la clase que implemente la interfaz dada y realizará las operaciones pertinentes, devolviendo el resultado.

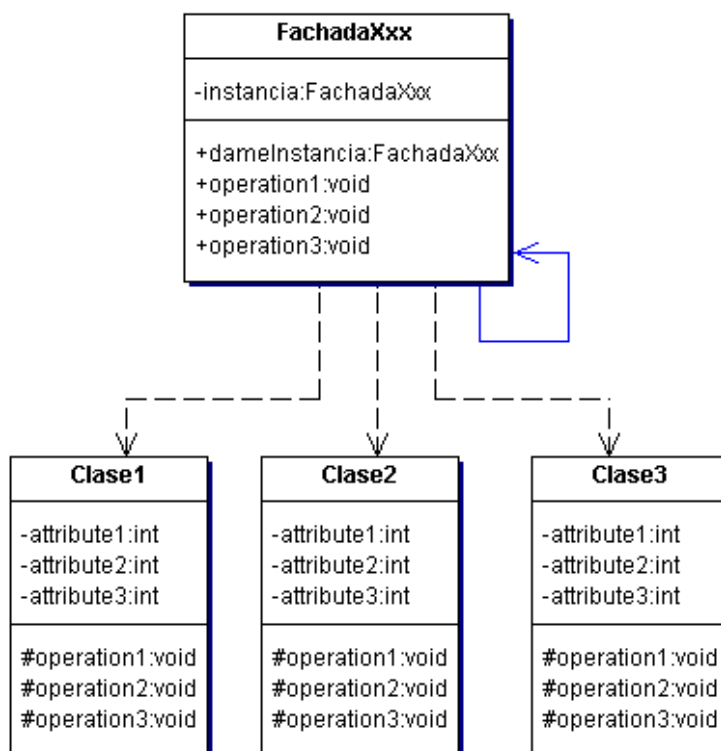
5.2. Diseño detallado

Para el diseño global de todo componente, se utilizará un meta-patrón genérico. El meta-patrón encapsula las funcionalidades de negocio de un componente a través de una única clase, especificando un diseño global del mismo.

El meta-patrón marca las bases, a través de dos patrones básicos, para el diseño de un componente de capa de negocio. La idea básica es encapsular todo el componente lógico de un componente, identificado por los distintos métodos públicos, a través de una única clase. Esta clase (*Patrón Fachada*) será la única con métodos públicos, siendo éstos los que propiamente ofrezcan al usuario la capa lógica. Por debajo de esta fachada, encontramos el componente “físico” donde se realizará la implementación real de los métodos ofrecidos por la fachada. Esta implementación se realizará por medio de distintas clases específicas del diseño, cuya declaración de métodos y clases será protegida y únicamente visible por los módulos del propio componente.

Además, la clase fachada que englobe el funcionamiento deberá implementar el *patrón Singleton* para poder ser accesible desde el exterior sin tener que ser instanciada.

El diseño genérico del meta-patrón viene dado por el siguiente diagrama de clases:



Este meta-patrón deberá tener unos cambios en las clases que usen *la librería betwixt*, ya que para que la librería funcione correctamente, algunos de los métodos deben ser públicos, entre ellos están los get y los sets de cada atributo, el constructor y dos métodos utilizados para añadir elementos a una lista de elementos, el acceso al iterator de la lista y el método que permite añadir objetos a la lista. Estas clases serán identificadas de forma que el nombre será: *nombreClaseXML*.



Además, cada una de estas clases deberá tener asociado un fichero .betwixt que tendrá el mismo nombre de la clase, en el que se indicarán las órdenes necesarias para el buen funcionamiento de la librería.

Una vez analizado el meta-patrón a utilizar, comenzaremos con el diseño de cada uno de los componentes.



5.2.1. Modelo.gestionTopicMap

Propósito

El componente gestionTopicMap implementa los siguientes requisitos software: RSFU-01, RSFU -02, RSFU -03, RSFU -04, RSFU -05, RSFU -06, RSFU -07, RSFU -08, RSFU -09, RSFU -10, RSFU-11, RSFU -12, RSFU -13, RSFU -14, RSFU -15, RSFU -16, RSFU -17, RSFU -18, RSFU -19, RSFU -20, RSFU -21, RSFU -22, RSFU -23, RSFU -24, RSFU -25, RSFU -26, RSFU -27, RSFU -28, RSFU -29, RSFU -30, RSFU -31, RSFU -32.

Dependencias

El componente de gestión analizado depende de la librería betwixt para poder guardar y rescatar los TM en formato xtm. Además, depende de los componentes gestionTopic, gestionAssociation y gestionScope, los cuales le permitirán implementar las funcionalidades del estándar de Topic Map.

Interfaces

El componente “ofrece” una interfaz de comunicación hacia el exterior (FachadaGestionTM), por medio de la cual los servicios prestados son accesibles para otros componentes del sistema. Los servicios están estructurados según el siguiente diagrama:



Proyecto Fin de Carrera-TMCreator



FachadaGestionTM

Propósito

El propósito de la clase FachadaGestionTM es implementar los requisitos software: RSFU-01, RSFU -02, RSFU -03, RSFU -04, RSFU -05, RSFU -06, RSFU -07, RSFU -08, RSFU -09, RSFU -10, RSFU-11, RSFU -12, RSFU -13, RSFU -14, RSFU -15, RSFU -16, RSFU -17, RSFU -18, RSFU -19, RSFU -20, RSFU -21, RSFU -22, RSFU -23, RSFU -24, RSFU -25, RSFU -26, RSFU -27, RSFU -28, RSFU -29 RSFU -30, RSFU -31, RSFU -32. Además, la clase hará de fachada para ofrecer a los demás componentes todas las funcionalidades del componente gestionTopicMap y evitar, de este modo, que las clases que formen al componente puedan ser vistas por los demás componentes.

Dependencias

La clase estudiada depende directamente de la clase TopicMapXML para llevar a cabo todas las funcionalidades que debe prestar al resto de componentes. Además, tiene dependencias externas con los componentes gestionTopic, gestionAssociation, gestionScope y persistencia.

Interfaces

La FachadaGestionTM tiene las siguientes operaciones de “acceso” al componente:

- *dameTopicMap()*
Descripción: inicializa el TopicMap con los datos recuperados del xtm.
Parámetros: void.
Return: el TopicMap inicializado.
- *dameTopics()*
Descripción: método que devuelve la lista de topics del TopicMap.
Parámetros: void.
Return: lista de topics.
- *dameNombreTopics()*
Descripción: método que devuelve una lista con los nombres de los topics.
Parámetros: void.
Return: lista con los nombres de los topics.
- *dameIdTopics()*
Descripción: método que devuelve los identificadores de los Topics del TM.
Parámetros: void.
Return: lista con los identificadores de los topics.
- *dameMetadatos()*
Descripción: método que devuelve los metadatos del Topic Map.
Parámetros: void.
Return: hasmap con los metadatos del topic map.
- *dameDatosTopic(String topic)*
Descripción: método que devuelve un hash con todos los datos del topic.



Parámetros:

- *topic*: id del topic.

Return: hash con los datos del topic.

- *dameInstanceOfTopic(String topic)*
Descripción: método que devuelve la lista de instancias del topic.
Parámetros:
 - *topic*: id del topic.Return: lista con las instancias del topic.
- *dameAsociaciones()*
Descripción: método que devuelve la lista de asociaciones del TopicMap.
Parámetros: void.
Return: lista de asociaciones.
- *dameDatosAsociacion(int numAsociacion)*
Descripción: método que devuelve un hash con todos los datos de la asociación.
Parámetros:
 - *numAsociacion*: índice de la asociación seleccionada.Return: hash con los datos de la asociación.
- *dameDatosMemberAsociacion(int asociacion, int member)*
Descripción: método que devuelve un hash con todos los datos del member de la asociación.
Parámetros:
 - *asociacion*: índice de la asociación seleccionada.
 - *Member*: índice del member seleccionado.Return: hash con los datos del member.
- *dameNombresAsociaciones()*
Descripción: método que devuelve una lista con los nombres de las asociaciones.
Parámetros: void.
Return: lista con los nombres de las asociaciones.
- *setTopicMap(TopicMapXML topicMap)*
Descripción: método que realiza el set del TopicMap
Parámetros:
 - *topicMap*: topicMap.Return: void.
- *crearTopic(String id, String baseName)*
Descripción: crea un nuevo Topic con las características deseadas.
Parámetros:
 - *id*: id del topic.
 - *baseName*: baseName del Topic creado.Return: boolean que permite saber si la operación se ha realizado correctamente.



- *removeTopic(String id)*
Descripción: elimina el Topic seleccionado del Topic Map.
Parámetros:
 - *id*: id del topic que se desea eliminar.Return: boolean que permite saber si la operación se ha realizado correctamente.
- *modificarTopic(String id, String baseName)*
Descripción: permite cambiar los datos informativos del Topic.
Parámetros:
 - *id*: id del topic que se desea modificar.
 - *baseName*: nuevo baseName del Topic.Return: boolean que permite saber si la operación se ha realizado correctamente.
- *crearInstanceOf(String tipo, String href, String idTopic)*
Descripción: crea un nuevo instanceOf a un Topic seleccionado.
Parámetros:
 - *tipo*: tipo de instancia (TopicRef o SubjectIndicatorRef).
 - *href*: referencia de la instancia.
 - *idTopic*: id del topic al que se desea añadir la nueva instancia.Return: boolean que permite saber si la operación se ha realizado correctamente.
- *removeInstanceOf(int indice, String idTopic)*
Descripción: elimina el elemento instanceOf elegido del Topic seleccionado.
Parámetros:
 - *indice*: índice de la instancia seleccionada.
 - *idTopic*: id del topic del que se borrará la instancia.Return: boolean que permite saber si la operación se ha realizado correctamente.
- *modificarInstanceOf(String tipo, int indice, String href, String idTopic)*
Descripción: modifica los datos del instanceOf elegido del Topic seleccionado.
Parámetros:
 - *tipo*: tipo de instancia (TopicRef o SubjectIndicatorRef).
 - *indice*: índice de la instancia seleccionada.
 - *href*: referencia de la instancia.
 - *idTopic*: id del topic del que se desea modificar la instancia.Return: boolean que permite saber si la operación se ha realizado correctamente.
- *addOccurrence(String idTopic, int tipoInstance, String instanceOf, int tipoResource, String resource, String idScope, ArrayList listaTopicRef, ArrayList listaResourceRef, ArrayList listaSubjectIndicatorRef)*
Descripción: añade una nueva occurrence a un Topic seleccionado.
Parámetros:
 - *idTopic*: id del topic al que se añadirá la occurrence.
 - *tipoInstance*: tipo de instanceOf (TopicRef o SubjectIndicatorRef)



- *instanceOf*: instanceOf del Occurrence.
- *tipoResource*: tipo de resource (ResourceRef o resourceData)
- *resource*: resource de la Occurrence.
- *idScope*: id del Scope de la Occurrence.
- *listaTopicRef*: lista de TopicRef del Scope asociado.
- *listaResourceRef*: lista de ResourceRef del Scope asociado.
- *listaSubjectIndicatorRef*: lista de SubjectIndicatorRef del Scope asociado.

Return: void

- *modificarOccurrence(String idTopic, int indice, int tipoInstance, String instanceOf, int tipoResource, String resource, String idScope, ArrayList listaTopicRef, ArrayList listaResourceRef, ArrayList listaSubjectIndicatorRef)*

Descripción: modifica los datos de la occurrence elegida del Topic seleccionado.

Parámetros:

- *idTopic*: id del topic del que se modificará la occurrence.
- *indice*: índice de la Occurrence seleccionada
- *tipoInstance*: tipo de instanteOf (TopicRef o SubjectIndicatorRef)
- *instanceOf*: instanceOf del Occurrence.
- *tipoResource*: tipo de resource (ResourceRef o resourceData)
- *resource*: resource de la Occurrence.
- *idScope*: id del Scope de la Occurrence.
- *listaTopicRef*: lista de TopicRef del Scope asociado.
- *listaResourceRef*: lista de ResourceRef del Scope asociado.
- *listaSubjectIndicatorRef*: lista de SubjectIndicatorRef del Scope asociado.

Return: void

- *removeOccurrence(String idTopic, int indice)*

Descripción: elimina la occurrence elegida del Topic seleccionado.

Parámetros:

- *idTopic*: id del topic seleccionado.
- *indice*: índice de la Occurrence seleccionada

Return: boolean que permite saber si la operación se ha realizado correctamente.

- *addBaseName(String idTopic, String idBaseName, String idScope, ArrayList listaTopicRef, ArrayList listaResourceRef, ArrayList listaSubjectIndicatorRef)*

Descripción: añade un nuevo BaseName a un Topic seleccionado.

Parámetros:

- *idTopic*: id del topic seleccionado.
- *idBaseName*: id del BaseName que se desea crear.
- *idScope*: id del Scope del BaseName.
- *listaTopicRef*: lista de TopicRef del Scope asociado.
- *listaResourceRef*: lista de ResourceRef del Scope asociado.
- *listaSubjectIndicatorRef*: lista de SubjectIndicatorRef del Scope asociado.

Return: void.

- *dameBaseName(idTopic)*

Descripción: método de acceso a los BaseNames del Topic.

Parámetros:



- *idTopic*: id del topic seleccionado.

Return: lista con los BaseNames del topic seleccionado.
- *removeBaseName(int indice, String idTopic)*

Descripción: elimina el BaseName elegido del Topic seleccionado.

Parámetros:

 - *indice*: índice del baseName seleccionado.
 - *idTopic*: id del topic seleccionado.

Return: boolean que permite saber si la operación se ha realizado correctamente.
- *dameDatosBaseName(String idTopic)*

Descripción: método que devuelve todos los datos de los baseNames asociados al topic seleccionado.

Parámetros:

 - *idTopic*: id del topic seleccionado.

Return: lista con los datos de los BaseNames del topic seleccionado.
- *modificarBaseName(String idTopic, int indice, String idBaseName, String idScope, ArrayList listaTopicRef, ArrayList listaResourceRef, ArrayList listaSubjectIndicatorRef)*

Descripción: modifica el BaseName elegido del Topic seleccionado.

Parámetros:

 - *idTopic*: id del topic seleccionado.
 - *indice*: índice del BaseName seleccionado
 - *idBaseName*: id del BaseName.
 - *idScope*: id del Scope del BaseName.
 - *listaTopicRef*: lista de TopicRef del Scope asociado.
 - *listaResourceRef*: lista de ResourceRef del Scope asociado.
 - *listaSubjectIndicatorRef*: lista de SubjectIndicatorRef del Scope asociado.

Return: void.
- *crearAsociation(String idAsociacion, String idMember, int tipoInstance, String href, String idScope, ArrayList listaTopicRef, ArrayList listaResourceRef, ArrayList listaSubjectIndicatorRef)*

Descripción: crea una nueva asociación.

Parámetros:

 - *idAsociacion*: id de la asociación.
 - *idMember*: id del member de la asociación.
 - *tipoInstance*: tipo de instanteOf (TopicRef o SubjectIndicatorRef).
 - *href*: referencia a la instancia.
 - *idScope*: id del Scope de la Occurrence.
 - *listaTopicRef*: lista de TopicRef del Scope asociado.
 - *listaResourceRef*: lista de ResourceRef del Scope asociado.
 - *listaSubjectIndicatorRef*: lista de SubjectIndicatorRef del Scope asociado.

Return: boolean que permite saber si la operación se ha realizado correctamente.
- *removeAsociation(int indice)*

Descripción: elimina la asociación seleccionada.

Parámetros:

- *indice*: índice de la asociación seleccionada.

Return: boolean que permite saber si la operación se ha realizado correctamente.

- *editarAssociation(int indice, String idAsociacion, String idMember, int tipoInstance, String href, String idScope, ArrayList listaTopicRef, ArrayList listaResourceRef, ArrayList listaSubjectIndicatorRef)*

Descripción: modifica los datos de la asociación.

Parámetros:

- *indice*: índice de la asociación seleccionada.
- *idAsociacion*: id de la asociación.
- *idMember*: id del member de la asociación.
- *tipoInstance*: tipo de instanteOf (TopicRef o SubjectIndicatorRef).
- *href*: referencia a la instancia.
- *idScope*: id del Scope de la Occurrence.
- *listaTopicRef*: lista de TopicRef del Scope asociado.
- *listaResourceRef*: lista de ResourceRef del Scope asociado.
- *listaSubjectIndicatorRef*: lista de SubjectIndicatorRef del Scope asociado.

Return: boolean que permite saber si la operación se ha realizado correctamente.

- *addMember(String indice, String idMember)*

Descripción: añade un nuevo miembro a la asociación.

Parámetros:

- *indice*: índice de la asociación seleccionada.
- *idMember*: id del nuevo member.

Return: boolean que permite saber si la operación se ha realizado correctamente.

- *removeMember(int indice, int indiceMember)*

Descripción: elimina el miembro seleccionado de la asociación.

Parámetros:

- *indice*: índice de la asociación seleccionada.
- *idMember*: id del nuevo member.

Return: boolean que permite saber si la operación se ha realizado correctamente.

- *dameMembers(int indice)*

Descripción: devuelve los miembros de la asociación.

Parámetros:

- *indice*: índice de la asociación seleccionada.

Return: lista de miembros de la asociación.

- *addTopicRefMember(String indice, String member, String topicRef)*

Descripción: crea un TopicRef al miembro de la asociación seleccionado.

Parámetros:

- *indice*: índice de la asociación seleccionada.



- *member*: id del member de la asociación.
 - *topicRef*: nuevo TopicRef.

Return: boolean que permite saber si la operación se ha realizado correctamente.
- *addResourceRefMember (String indice, String member, String resourceRef)*

Descripción: crea un ResourceRef al miembro de la asociación seleccionado.

Parámetros:

 - *indice*: índice de la asociación seleccionada.
 - *member*: id del member de la asociación.
 - *resourceRef*: nuevo ResourceRef.

Return: boolean que permite saber si la operación se ha realizado correctamente.
- *addSubjectIndicatorRefMember (String indice, String member, String subjectIndicatorRef)*

Descripción: crea un SubjectIndicatorRef al miembro de la asociación seleccionado.

Parámetros:

 - *indice*: índice de la asociación seleccionada.
 - *member*: id del member de la asociación.
 - *subjectIndicatorRef*: nuevo SubjectIndicatorRef.

Return: boolean que permite saber si la operación se ha realizado correctamente.
- *removeTopicRefMember (String indice, String member, String topicRef)*

Descripción: elimina el TopicRef elegido del miembro de la asociación.

Parámetros:

 - *indice*: índice de la asociación seleccionada.
 - *member*: id del member de la asociación.
 - *topicRef*: elemento que se desea eliminar del miembro de la asociación.

Return: boolean que permite saber si la operación se ha realizado correctamente.
- *removeResourceRefMember (String indice, String member, String resourceRef)*

Descripción: elimina el ResourceRef elegido del miembro de la asociación.

Parámetros:

 - *indice*: índice de la asociación seleccionada.
 - *member*: id del member de la asociación.
 - *resourceRef*: elemento que se desea eliminar del miembro de la asociación.

Return: boolean que permite saber si la operación se ha realizado correctamente.
- *removeSubjectIndicatorRefMember (String indice, String member, String subjectIndicatorRef)*

Descripción: elimina el SubjectIndicatorRef elegido del miembro de la asociación.

Parámetros:

 - *indice*: índice de la asociación seleccionada.
 - *member*: id del member de la asociación.

- *subjectIndicatorRef*: elemento que se desea eliminar del miembro de la asociación.
- Return: boolean que permite saber si la operación se ha realizado correctamente.
- *dameNombreScopes()*
 - Descripción: devuelve lista con los nombres de los scopes del TopicMap.
 - Parámetros: void.
 - Return: lista con los nombres de los Scopes.
- *dameTopicsScope(String scopeSelected)*
 - Descripción: devuelve los Topics que son del Scope seleccionado.
 - Parámetros:
 - *scopeSelected*: scope seleccionado.
 - Return: lista de topics del scope.
- *dameAsociacionesScope (String scopeSelected)*
 - Descripción: devuelve las asociaciones que son del Scope seleccionado.
 - Parámetros:
 - *scopeSelected*: scope seleccionado.
 - Return: lista de asociaciones del scope.
- *dameOccurrenceTopic(String idTopic)*
 - Descripción: devuelve las occurrences del topic seleccionado.
 - Parámetros:
 - *idTopic*: topic seleccionado.
 - Return: lista de las occurrences del topic seleccionado.
- *dameDatosOccurrenceTopic(String idTopic)*
 - Descripción: devuelve los datos de las occurrences del topic seleccionado.
 - Parámetros:
 - *idTopic*: topic seleccionado.
 - Return: lista de los datos de las occurrences del topic seleccionado.
- *modificarSubjectIdentity(String idTopic, String resourceRef)*
 - Descripción: modifica los datos del SubjectIdentity seleccionado.
 - Parámetros:
 - *idTopic*: topic seleccionado.
 - *resourceRef*: nuevos datos del SubjectIdentity
 - Return: boolean que permite saber si la operación se ha realizado correctamente.
- *addTopicRefSubjectIdentity(String idTopic, String href)*
 - Descripción: añade un TopicRef al SubjectIdentity.
 - Parámetros:
 - *idTopic*: topic seleccionado.
 - *href*: referencia del nuevo TopicRef
 - Return: boolean que permite saber si la operación se ha realizado correctamente.



- *addSubjectIndicatorSubjectIdentity (String idTopic, String href)*
Descripción: añade un SubjectIndicator al SubjectIdentity.
Parámetros:
 - *idTopic*: topic seleccionado.
 - *href*: referencia del nuevo SubjectIndicatorReturn: boolean que permite saber si la operación se ha realizado correctamente.
- *removeTopicRefSubjectIdentity(int indice, String idTopic)*
Descripción: elimina el TopicRef del SubjectIdentity.
Parámetros:
 - *indice*: índice del TopicRef seleccionado.
 - *idTopic*: topic seleccionado.Return: boolean que permite saber si la operación se ha realizado correctamente.
- *removeSubjectIndicatorSubjectIdentity (int indice, String idTopic)*
Descripción: elimina el SubjectIndicator del SubjectIdentity.
Parámetros:
 - *indice*: índice del SubjectIndicator seleccionado.
 - *idTopic*: topic seleccionado.Return: boolean que permite saber si la operación se ha realizado correctamente.
- *dameDatosModificarAsociacion(int indiceAsociacion)*
Descripción: devuelve los datos de modificación de la asociación.
Parámetros:
 - *indiceAsociacion*: índice de la asociación seleccionada.Return: hash con los datos de la asociación.
- *modificarRolSpecMember(String asociacion, String member, String rolSpec, String tipoRol)*
Descripción: modifica el rol del miembro de la asociación seleccionado.
Parámetros:
 - *asociacion*: índice de la asociación seleccionada.
 - *member*: índice del miembro de la asociación seleccionado.
 - *rolSpec*: nuevo rol del miembro de la asociación.
 - *tipoRol*: tipo del elemento del rol (TopicRef oSubjectIndicatorRef).Return: boolean que permite saber si la operación se ha realizado correctamente.
- *crearTM()*
Descripción: método que permite al usuario crear un nuevo Topic Map.
Parámetros: void.
Return: void.
- *cargarTM(String path)*
Descripción: carga los datos de un fichero xtm.



Parámetros:

- *path*: path del fichero xtm que se va a cargar.

Return: string con los datos del xml cargados.

- *toObjeto (String xml)*
Descripción: transforma el xml recuperado en objetos.
Parámetros:
 - *xml*: string con el xml cargado.Return: TopicMapXML con los datos.
- *toXML(TopicMapXML topicMap)*
Descripción: método que transforma los objetos en xml.
Parámetros:
 - *topicMap*: los datos que deben transformarse a xml.Return: el xml con los datos.
- *guardar(String xml)*
Descripción: guarda el xml en un fichero físico.
Parámetros:
 - *xml*: los datos en formato xml.Return: void.
- *guardaComo(String path,String xml)*
Descripción: guarda el xml en un fichero físico posibilitando el cambio de nombre del fichero.
Parámetros:
 - *path*: path en el que se guardará el fichero xtm.
 - *xml*: los datos en formato xml.Return: void.
- *separaMetadatos()*
Descripción: identifica los metadatos dentro de la lista de Topics.
Parámetros: void.
Return: void.
- *dameMetadatos()*
Descripción: obtiene los metadatos del Topic Map.
Parámetros: void
Return: hashmap con los metadatos.
- *setMetadatos(HashMap datos)*
Descripción: guarda los metadatos del TopicMap.
Parámetros:
 - *datos*: los metadatos del Topic Map.Return: void.



- *getNombreId(String id)*
Descripción: obtiene el nombre a partir del identificador del Topic.
Parámetros:
 - *id*: identificador del Topic.Return: base name del Topic.



TopicMapXML

Propósito

El propósito de la clase TopicMap es implementar los requisitos software: RSFU-01, RSFU -02, RSFU -03, RSFU -04.

Dependencias

No tiene dependencias internas. Tiene dependencias externas con la librería betwixt, que permite grabar y recuperar la información en formato XML.

Interfaces

La clase TopicMapXML tiene las siguientes operaciones, los get y los set de los atributos son métodos que no se describirán:

- *ponTopics(LinkedList topic)*
Descripción: método que permite añadir una lista de Topics
Parámetros
 - *topic*: nueva lista de Topics.Return: void.
- *ponAsociaciones(LinkedList association)*
Descripción: método que permite añadir una lista de asociaciones.
Parámetros
 - *association*: nueva lista de asociaciones.Return: void.
- *addTopic(TopicXML topic)*
Descripción: método que permite al usuario añadir un nuevo Topic.
Parámetros
 - *topic*: nuevo topicReturn: void.
- *getTopicIterator()*
Descripción: método que accede al iterator de los topic.
Parámetros: void
Return: iterator de la lista de topic.
- *addAssociation(AssociationXML asociacion)*
Descripción: método que permite al usuario añadir una nueva asociación.
Parámetros:
 - *asociacion*: nueva asociación.Return: void.
- *getAssociationIterator()*
Descripción: método que accede al iterator de las asociaciones.
Parámetros: void
Return: iterator de la lista de asociaciones.



5.2.2. Modelo.gestionTopic

Propósito

El componente gestionTopic implementa los siguientes requisitos software: RSFU -05, RSFU -06, RSFU -07, RSFU -08, RSFU -09, RSFU -10, RSFU-11, RSFU -12, RSFU -13, RSFU -14, RSFU -15, RSFU -16, RSFU -17, RSFU -18, RSFU -32.

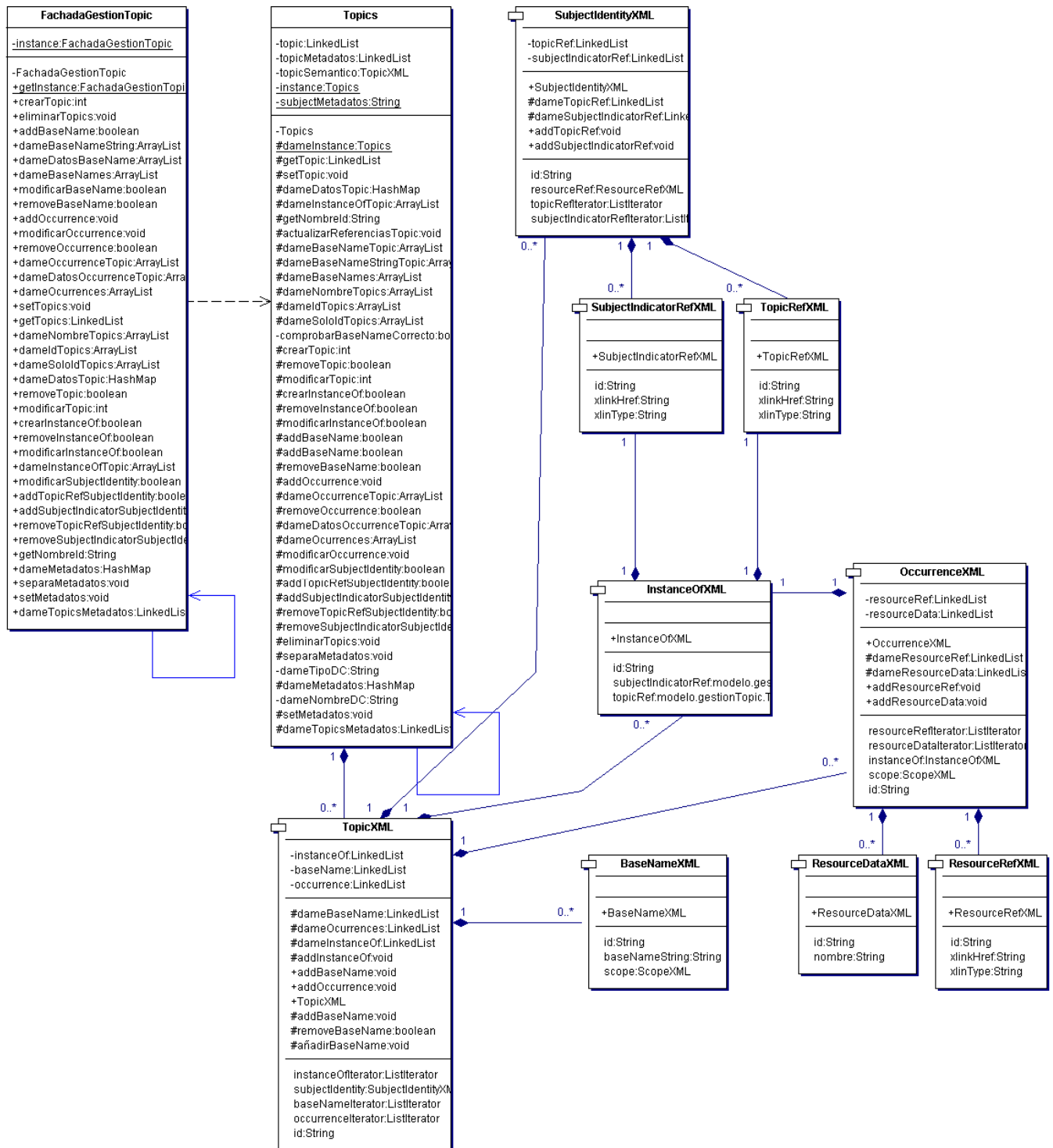
Dependencias

El componente gestionTopic depende de la librería betwixt para recuperar y grabar la información en formato xtm.

Interfaces

El componente “ofrece” una interfaz de comunicación hacia el exterior (FachadaGestionTopic), por medio de la cual los servicios prestados son accesibles para otros componentes del sistema. Los servicios están estructurados según el siguiente diagrama:

(Diagrama en la siguiente página)





FachadaGestionTopic

Propósito

El propósito de la clase FachadaGestionTopic es implementar los requisitos software: RSFU -04, RSFU -05, RSFU -06, RSFU -07, RSFU -08, RSFU -09, RSFU -10, RSFU-11, RSFU -12, RSFU -13, RSFU -14, RSFU -15, RSFU -16, RSFU -17, RSFU -18, RSFU -19, RSFU -20, RSFU -21, RSFU -22, RSFU -32. Además, la clase hará de fachada para ofrecer a los demás componentes todas las funcionalidades del componente gestionTopic.

Dependencias

La clase estudiada tiene dependencias internas con la clase Topics.

Interfaces

La FachadaGestionTopic tiene las siguientes operaciones de “acceso” al componente:

- *crearTopic(String id, String baseName)*
Descripción: crea un nuevo Topic con las características deseadas.
Parámetros:
 - *id*: id del topic.
 - *baseName*: baseName del Topic creado.Return: boolean que permite saber si la operación se ha realizado correctamente.
- *eliminarTopics()*
Descripción: elimina todos los topics.
Parámetros: void
Return: void.
- *addBaseName(String idTopic, String idBaseName, String idScope, ArrayList listaTopicRef, ArrayList listaResourceRef, ArrayList listaSubjectIndicatorRef)*
Descripción: añade un nuevo BaseName a un Topic seleccionado.
Parámetros:
 - *idTopic*: id del topic seleccionado.
 - *idBaseName*: id del BaseName que se desea crear.
 - *idScope*: id del Scope del BaseName.
 - *listaTopicRef*: lista de TopicRef del Scope asociado.
 - *listaResourceRef*: lista de ResourceRef del Scope asociado.
 - *listaSubjectIndicatorRef*: lista de SubjectIndicatorRef del Scope asociado.Return: void.
- *dameBaseNames(idTopic)*
Descripción: método de acceso a los BaseNames del Topic.
Parámetros:
 - *idTopic*: id del topic seleccionado.Return: lista con los BaseNames del topic seleccionado.
- *dameBaseNameString(idTopic)*

Descripción: método de acceso a los BaseNames String de todos los BaseName del Topic.

Parámetros:

- *idTopic*: id del topic seleccionado.

Return: lista con los BaseNames del topic seleccionado.

- *removeBaseName(int indice, String idTopic)*

Descripción: elimina el BaseName elegido del Topic seleccionado.

Parámetros:

- *indice*: índice del baseName seleccionado.
- *idTopic*: id del topic seleccionado.

Return: boolean que permite saber si la operación se ha realizado correctamente.

- *dameDatosBaseName(String idTopic)*

Descripción: método que devuelve todos los datos de los baseNames asociados al topic seleccionado.

Parámetros:

- *idTopic*: id del topic seleccionado.

Return: lista con los datos de los BaseNames del topic seleccionado.

- *modificarBaseName(String idTopic, int indice, String idBaseName, String idScope, ArrayList listaTopicRef, ArrayList listaResourceRef, ArrayList listaSubjectIndicatorRef)*

Descripción: modifica el BaseName elegido del Topic seleccionado.

Parámetros:

- *idTopic*: id del topic seleccionado.
- *indice*: índice del BaseName seleccionado
- *idBaseName*: id del BaseName.
- *idScope*: id del Scope del BaseName.
- *listaTopicRef*: lista de TopicRef del Scope asociado.
- *listaResourceRef*: lista de ResourceRef del Scope asociado.
- *listaSubjectIndicatorRef*: lista de SubjectIndicatorRef del Scope asociado.

Return: void.

- *addOccurrence(String idTopic, int tipoInstance, String instanceOf, int tipoResource, String resource, String idScope, ArrayList listaTopicRef, ArrayList listaResourceRef, ArrayList listaSubjectIndicatorRef)*

Descripción: añade una nueva occurrence a un Topic seleccionado.

Parámetros:

- *idTopic*: id del topic al que se añadirá la occurrence.
- *tipoInstance*: tipo de instanteOf (TopicRef o SubjectIndicatorRef)
- *instanceOf*: instanceOf del Occurrence.
- *tipoResource*: tipo de resource (ResourceRef o resourceData)
- *resource*: resource de la Occurrence.
- *idScope*: id del Scope de la Occurrence.
- *listaTopicRef*: lista de TopicRef del Scope asociado.
- *listaResourceRef*: lista de ResourceRef del Scope asociado.
- *listaSubjectIndicatorRef*: lista de SubjectIndicatorRef del Scope asociado.

Return: void

- *modificarOccurrence(String idTopic, int indice, int tipoInstance, String instanceOf, int tipoResource, String resource, String idScope, ArrayList listaTopicRef, ArrayList listaResourceRef, ArrayList listaSubjectIndicatorRef)*

Descripción: modifica los datos de la occurrence elegida del Topic seleccionado.

Parámetros:

- *idTopic*: id del topic del que se modificará la occurrence.
- *indice*: índice de la Occurrence seleccionada
- *tipoInstance*: tipo de instanceOf (TopicRef o SubjectIndicatorRef)
- *instanceOf*: instanceOf del Occurrence.
- *tipoResource*: tipo de resource (ResourceRef o resourceData)
- *resource*: resource de la Occurrence.
- *idScope*: id del Scope de la Occurrence.
- *listaTopicRef*: lista de TopicRef del Scope asociado.
- *listaResourceRef*: lista de ResourceRef del Scope asociado.
- *listaSubjectIndicatorRef*: lista de SubjectIndicatorRef del Scope asociado.

Return: void

- *removeOccurrence(String idTopic, int indice)*

Descripción: elimina la occurrence elegida del Topic seleccionado.

Parámetros:

- *idTopic*: id del topic seleccionado.
- *indice*: índice de la Occurrence seleccionada

Return: boolean que permite saber si la operación se ha realizado correctamente.

- *dameOccurrenceTopic(String idTopic)*

Descripción: devuelve las occurrences del topic seleccionado.

Parámetros:

- *idTopic*: topic seleccionado.

Return: lista de las occurrences del topic seleccionado.

- *dameOccurrences(String idTopic)*

Descripción: devuelve las occurrences del topic en objetos OccurrenceXML.

Parámetros:

- *idTopic*: topic seleccionado.

Return: lista de las occurrences del topic seleccionado.

- *dameDatosOccurrenceTopic(String idTopic)*

Descripción: devuelve los datos de las occurrences del topic seleccionado.

Parámetros:

- *idTopic*: topic seleccionado.

Return: lista de los datos de las occurrences del topic seleccionado.

- *dameTopics()*

Descripción: método que devuelve la lista de topics del TopicMap.

Parámetros: void.

Return: lista de topics.

- *dameDatosTopic(String topic)*
Descripción: método que devuelve un hash con todos los datos del topic.
Parámetros:
 - *topic*: id del topic.Return: hash con los datos del topic.
- *dameNombreTopics()*
Descripción: método que devuelve una lista con los nombres de los topics.
Parámetros: void.
Return: lista con los nombres de los topics.
- *removeTopic(String id)*
Descripción: elimina el Topic seleccionado del Topic Map.
Parámetros:
 - *id*: id del topic que se desea eliminar.Return: boolean que permite saber si la operación se ha realizado correctamente.
- *modificarTopic(String id, String baseName)*
Descripción: permite cambiar los datos informativos del Topic.
Parámetros:
 - *id*: id del topic que se desea modificar.
 - *baseName*: nuevo baseName del Topic.Return: boolean que permite saber si la operación se ha realizado correctamente.
- *crearInstanceOf(String tipo, String href, String idTopic)*
Descripción: crea un nuevo instanceOf a un Topic seleccionado.
Parámetros:
 - *tipo*: tipo de instancia (TopicRef o SubjectIndicatorRef).
 - *href*: referencia de la instancia.
 - *idTopic*: id del topic al que se desea añadir la nueva instancia.Return: boolean que permite saber si la operación se ha realizado correctamente.
- *removeInstanceOf(int indice, String idTopic)*
Descripción: elimina el instanceOf elegido del Topic seleccionado.
Parámetros:
 - *indice*: índice de la instancia seleccionada.
 - *idTopic*: id del topic del que se borrará la instancia.Return: boolean que permite saber si la operación se ha realizado correctamente.
- *modificarInstanceOf(String tipo, int indice, String href, String idTopic)*
Descripción: modifica los datos del instanceOf elegido del Topic seleccionado.
Parámetros:
 - *tipo*: tipo de instancia (TopicRef o SubjectIndicatorRef).

- *indice*: índice de la instancia seleccionada.
 - *href*: referencia de la instancia.
 - *idTopic*: id del topic del que se desea modificar la instancia.

Return: boolean que permite saber si la operación se ha realizado correctamente.
- *dameInstanceOfTopic(String topic)*

Descripción: método que devuelve la lista de instancias del topic.

Parámetros:

 - *topic*: id del topic.

Return: lista con las instancias del topic.
- *modificarSubjectIdentity(String idTopic, String resourceRef)*

Descripción: modifica los datos del SubjectIdentity seleccionado.

Parámetros:

 - *idTopic*: topic seleccionado.
 - *resourceRef*: nuevos datos del SubjectIdentity

Return: boolean que permite saber si la operación se ha realizado correctamente.
- *addTopicRefSubjectIdentity(String idTopic, String href)*

Descripción: añade un TopicRef al SubjectIdentity.

Parámetros:

 - *idTopic*: topic seleccionado.
 - *href*: referencia del nuevo TopicRef

Return: boolean que permite saber si la operación se ha realizado correctamente.
- *addSubjectIndicatorSubjectIdentity (String idTopic, String href)*

Descripción: añade un SubjectIndicator al SubjectIdentity.

Parámetros:

 - *idTopic*: topic seleccionado.
 - *href*: referencia del nuevo SubjectIndicator

Return: boolean que permite saber si la operación se ha realizado correctamente.
- *removeTopicRefSubjectIdentity(int indice, String idTopic)*

Descripción: elimina el TopicRef del SubjectIdentity.

Parámetros:

 - *indice*: índice del TopicRef seleccionado.
 - *idTopic*: topic seleccionado.

Return: boolean que permite saber si la operación se ha realizado correctamente.
- *removeSubjectIndicatorSubjectIdentity (int indice, String idTopic)*

Descripción: elimina el SubjectIndicator del SubjectIdentity.

Parámetros:

 - *indice*: índice del SubjectIndicator seleccionado.
 - *idTopic*: topic seleccionado.



Return: boolean que permite saber si la operación se ha realizado correctamente.

- *separaMetadatos()*
Descripción: identifica los metadatos dentro de la lista de Topics.
Parámetros: void.
Return: void.
- *setMetadatos(HashMap datos)*
Descripción: guarda los metadatos del TopicMap.
Parámetros:
 - *datos*: los metadatos del Topic Map.Return: void.
- *getNombreId(String id)*
Descripción: obtiene el nombre a partir del identificador del Topic.
Parámetros:
 - *id*: identificador del Topic.Return: base name del Topic.
- *dameMetadatos()*
Descripción: obtiene los metadatos del Topic Map.
Parámetros: void
Return: hashmap con los metadatos.
- *dameTopicsMetadatos()*
Descripción: obtiene los metadatos como Topics del Topic Map.
Parámetros: void
Return: lista de topics con los metadatos.
- *getInstance();*
Descripción: método que permite al usuario acceder al componente. Gracias a este método, se implementa el patrón Singleton.
Parámetros: no tiene.



Topics

Propósito

El propósito de la clase Topic es implementar los requisitos software RSFU -04, RSFU -05, RSFU -06, RSFU -07, RSFU -08, RSFU -09, RSFU -10, RSFU -11, RSFU -12, RSFU -13, RSFU -14, RSFU -15, RSFU -16, RSFU -17, RSFU -18, RSFU -19, RSFU -20, RSFU -21, RSFU -22, RSFU -32.

Dependencias

La clase estudiada depende directamente de las clases TopicXML.

Interfaces

La clase Topics dispone de las siguientes operaciones:

- *crearTopic(String id, String baseName)*
Descripción: crea un nuevo Topic con las características deseadas.
Parámetros:
 - *id*: id del topic.
 - *baseName*: baseName del Topic creado.Return: boolean que permite saber si la operación se ha realizado correctamente.
- *addBaseName(String idTopic, String idBaseName, String idScope, ArrayList listaTopicRef, ArrayList listaResourceRef, ArrayList listaSubjectIndicatorRef)*
Descripción: añade un nuevo BaseName a un Topic seleccionado.
Parámetros:
 - *idTopic*: id del topic seleccionado.
 - *idBaseName*: id del BaseName que se desea crear.
 - *idScope*: id del Scope del BaseName.
 - *listaTopicRef*: lista de TopicRef del Scope asociado.
 - *listaResourceRef*: lista de ResourceRef del Scope asociado.
 - *listaSubjectIndicatorRef*: lista de SubjectIndicatorRef del Scope asociado.Return: void.
- *dameBaseNames(idTopic)*
Descripción: método de acceso a los BaseNames del Topic.
Parámetros:
 - *idTopic*: id del topic seleccionado.Return: lista con los BaseNames del topic seleccionado.
- *dameBaseNameStringTopic(idTopic)*
Descripción: método de acceso a los BaseNames String de todos los BaseName del Topic.
Parámetros:
 - *idTopic*: id del topic seleccionado.Return: lista con los BaseNames del topic seleccionado.

- *removeBaseName(int indice, String idTopic)*
Descripción: elimina el BaseName elegido del Topic seleccionado.
Parámetros:
 - *indice*: índice del baseName seleccionado.
 - *idTopic*: id del topic seleccionado.Return: boolean que permite saber si la operación se ha realizado correctamente.
- *dameBaseNameTopic (String idTopic)*
Descripción: método que devuelve todos los datos de los baseNames asociados al topic seleccionado.
Parámetros:
 - *idTopic*: id del topic seleccionado.Return: lista con los datos de los BaseNames del topic seleccionado.
- *addBaseName (String idTopic, int indice, String idBaseName, String idScope, ArrayList listaTopicRef, ArrayList listaResourceRef, ArrayList listaSubjectIndicatorRef)*
Descripción: modifica el BaseName elegido del Topic seleccionado.
Parámetros:
 - *idTopic*: id del topic seleccionado.
 - *indice*: índice del BaseName seleccionado
 - *idBaseName*: id del BaseName.
 - *idScope*: id del Scope del BaseName.
 - *listaTopicRef*: lista de TopicRef del Scope asociado.
 - *listaResourceRef*: lista de ResourceRef del Scope asociado.
 - *listaSubjectIndicatorRef*: lista de SubjectIndicatorRef del Scope asociado.Return: void.
- *addOccurrence(String idTopic, int tipoInstance, String instanceOf, int tipoResource, String resource, String idScope, ArrayList listaTopicRef, ArrayList listaResourceRef, ArrayList listaSubjectIndicatorRef)*
Descripción: añade una nueva occurrence a un Topic seleccionado.
Parámetros:
 - *idTopic*: id del topic al que se añadirá la occurrence.
 - *tipoInstance*: tipo de instanteOf (TopicRef o SubjectIndicatorRef)
 - *instanceOf*: instanceOf del Occurrence.
 - *tipoResource*: tipo de resource (ResourceRef o resourceData)
 - *resource*: resource de la Occurrence.
 - *idScope*: id del Scope de la Occurrence.
 - *listaTopicRef*: lista de TopicRef del Scope asociado.
 - *listaResourceRef*: lista de ResourceRef del Scope asociado.
 - *listaSubjectIndicatorRef*: lista de SubjectIndicatorRef del Scope asociado.Return: void
- *modificarOccurrence(String idTopic, int indice, int tipoInstance, String instanceOf, int tipoResource, String resource, String idScope, ArrayList listaTopicRef, ArrayList listaResourceRef, ArrayList listaSubjectIndicatorRef)*
Descripción: modifica los datos de la occurrence elegida del Topic seleccionado.

Parámetros:

- *idTopic*: id del topic del que se modificará la occurrence.
- *indice*: índice de la Occurrence seleccionada
- *tipoInstance*: tipo de instanceOf (TopicRef o SubjectIndicatorRef)
- *instanceOf*: instanceOf del Occurrence.
- *tipoResource*: tipo de resource (ResourceRef o resourceData)
- *resource*: resource de la Occurrence.
- *idScope*: id del Scope de la Occurrence.
- *listaTopicRef*: lista de TopicRef del Scope asociado.
- *listaResourceRef*: lista de ResourceRef del Scope asociado.
- *listaSubjectIndicatorRef*: lista de SubjectIndicatorRef del Scope asociado.

Return: void

- *removeOccurrence(String idTopic,int indice)*
Descripción: elimina la occurrence elegida del Topic seleccionado.
Parámetros:
 - *idTopic*: id del topic seleccionado.
 - *indice*: índice de la Occurrence seleccionadaReturn: boolean que permite saber si la operación se ha realizado correctamente.
- *dameOccurrenceTopic(String idTopic)*
Descripción: devuelve las occurrences del topic seleccionado.
Parámetros:
 - *idTopic*: topic seleccionado.Return: lista de las occurrences del topic seleccionado.
- *dameOccurrences(String idTopic)*
Descripción: devuelve las occurrences del topic en objetos OccurrenceXML.
Parámetros:
 - *idTopic*: topic seleccionado.Return: lista de las occurrences del topic seleccionado.
- *dameDatosOccurrenceTopic(String idTopic)*
Descripción: devuelve los datos de las occurrences del topic seleccionado en HashMap.
Parámetros:
 - *idTopic*: topic seleccionado.Return: lista de los datos de las occurrences del topic seleccionado.
- *dameDatosTopic(String topic)*
Descripción: método que devuelve un hash con todos los datos del topic.
Parámetros:
 - *topic*: id del topic.Return: hash con los datos del topic.
- *dameNombreTopics()*
Descripción: método que devuelve una lista con los nombres de los topics.
Parámetros: void.

Return: lista con los nombres de los topics.

- *removeTopic(String id)*
Descripción: elimina el Topic seleccionado del Topic Map.
Parámetros:
 - *id*: id del topic que se desea eliminar.Return: boolean que permite saber si la operación se ha realizado correctamente.
- *modificarTopic(String id, String baseName)*
Descripción: permite cambiar los datos informativos del Topic.
Parámetros:
 - *id*: id del topic que se desea modificar.
 - *baseName*: nuevo baseName del Topic.Return: boolean que permite saber si la operación se ha realizado correctamente.
- *crearInstanceOf(String tipo, String href, String idTopic)*
Descripción: crea un nuevo instanceOf a un Topic seleccionado.
Parámetros:
 - *tipo*: tipo de instancia (TopicRef o SubjectIndicatorRef).
 - *href*: referencia de la instancia.
 - *idTopic*: id del topic al que se desea añadir la nueva instancia.Return: boolean que permite saber si la operación se ha realizado correctamente.
- *removeInstanceOf(int indice, String idTopic)*
Descripción: elimina el instanceOf elegido del Topic seleccionado.
Parámetros:
 - *indice*: índice de la instancia seleccionada.
 - *idTopic*: id del topic del que se borrará la instancia.Return: boolean que permite saber si la operación se ha realizado correctamente.
- *modificarInstanceOf(String tipo, int indice, String href, String idTopic)*
Descripción: modifica los datos del instanceOf elegido del Topic seleccionado.
Parámetros:
 - *tipo*: tipo de instancia (TopicRef o SubjectIndicatorRef).
 - *indice*: índice de la instancia seleccionada.
 - *href*: referencia de la instancia.
 - *idTopic*: id del topic del que se desea modificar la instancia.Return: boolean que permite saber si la operación se ha realizado correctamente.
- *dameInstanceOfTopic(String topic)*
Descripción: método que devuelve la lista de instancias del topic.
Parámetros:
 - *topic*: id del topic.Return: lista con las instancias del topic.

- *modificarSubjectIdentity(String idTopic, String resourceRef)*
Descripción: modifica los datos del SubjectIdentity seleccionado.
Parámetros:
 - *idTopic*: topic seleccionado.
 - *resourceRef*: nuevos datos del SubjectIdentityReturn: boolean que permite saber si la operación se ha realizado correctamente.
- *addTopicRefSubjectIdentity(String idTopic, String href)*
Descripción: añade un TopicRef al SubjectIdentity.
Parámetros:
 - *idTopic*: topic seleccionado.
 - *href*: referencia del nuevo TopicRefReturn: boolean que permite saber si la operación se ha realizado correctamente.
- *addSubjectIndicatorSubjectIdentity (String idTopic, String href)*
Descripción: añade un SubjectIndicator al SubjectIdentity.
Parámetros:
 - *idTopic*: topic seleccionado.
 - *href*: referencia del nuevo SubjectIndicatorReturn: boolean que permite saber si la operación se ha realizado correctamente.
- *removeTopicRefSubjectIdentity(int indice, String idTopic)*
Descripción: elimina el TopicRef del SubjectIdentity.
Parámetros:
 - *indice*: índice del TopicRef seleccionado.
 - *idTopic*: topic seleccionado.Return: boolean que permite saber si la operación se ha realizado correctamente.
- *removeSubjectIndicatorSubjectIdentity (int indice, String idTopic)*
Descripción: elimina el SubjectIndicator del SubjectIdentity.
Parámetros:
 - *indice*: índice del SubjectIndicator seleccionado.
 - *idTopic*: topic seleccionado.Return: boolean que permite saber si la operación se ha realizado correctamente.
- *separaMetadatos()*
Descripción: identifica los metadatos dentro de la lista de Topics.
Parámetros: void.
Return: void.
- *setMetadatos(HashMap datos)*
Descripción: guarda los metadatos del TopicMap.
Parámetros:



- *datos*: los metadatos del Topic Map.
Return: void.
- *getNombreId(String id)*
Descripción: obtiene el nombre a partir del identificador del Topic.
Parámetros:
 - *id*: identificador del Topic.Return: base name del Topic.
- *dameMetadatos()*
Descripción: obtiene los metadatos del Topic Map.
Parámetros: void
Return: hashmap con los metadatos.
- *dameTopicsMetadatos()*
Descripción: obtiene los metadatos como Topics del Topic Map.
Parámetros: void
Return: lista de topics con los metadatos.
- *dameInstance();*
Descripción: método que permite al usuario acceder al componente. Gracias a este método se implementa el patrón singleton.
Parámetros: void
Return: la única instancia del objeto.

TopicXML

Propósito

El propósito de la clase BaseName es implementar los requisitos software: RSFU -19, RSFU -20, RSFU -21, RSFU -22.

Dependencias

Tiene dependencias internas con las clases OccurrenceXML, BaseNameXML, InstanceOfXML y SubjectIdentityXML. Tiene dependencias externas con la librería betwixt, que permite grabar y recuperar la información en formato XML.

Interfaces

La clase TopicXML tiene las siguientes operaciones. Los get y los set de los atributos son métodos que no se describirán:

- *dameBaseName()*
Descripción: método de acceso a los BaseNames del Topic.
Parámetros: void
Return: lista con los BaseNames del topic seleccionado.
- *dameOcurrrences()*
Descripción: método de acceso a los Occurrences del Topic.
Parámetros: void
Return: lista con los Occurrence del topic seleccionado.
- *dameInstanceOf()*
Descripción: método de acceso los InstanceOf del Topic.
Parámetros: void
Return: lista con los InstanceOf del topic seleccionado
- *addInstanceOf(InstanceOfXML instanceOf)*
Descripción: método que añade un nuevo InstanceOf al Topic.
Parámetros:
 - *instanceOf*: nueva instancia del Topic.Return: void
- *getInstanceOflterator()*
Descripción: accedente al iterador de la lista de instancias del Topic.
Parámetros: void
Return: el iterador de la lista
- *addBaseName(BaseNameXML baseName)*
Descripción: método que añade un nuevo BaseName al Topic.
Parámetros:
 - *baseName*: nuevo BaseName del Topic.Return: void



- *getBaseNameIterator()*
Descripción: accedente al iterador de la lista de BaseNames del Topic.
Parámetros: void
Return: el iterador de la lista.
- *addOccurrence(OccurrenceXML occurrence)*
Descripción: método que añade una nueva Occurrence al Topic.
Parámetros:
 - *occurrence*: nueva Occurrence del Topic.Return: void
- *getOccurrenceIterator()*
Descripción: accedente al iterador de la lista de Occurrences del Topic.
Parámetros: void
Return: el iterador de la lista.
- *addBaseName(String nameString)*
Descripción: método que añade una nueva BaseName a partir de su nameString.
Parámetros:
 - *nameString*: nombre de la nueva BaseName.Return: void
- *removeBaseName(String idBaseName)*
Descripción: método de una BaseName del Topic.
Parámetros:
 - *idBaseName*: id del BaseName que se desea borrar.Return: void
- *añadirBaseName(String nombre,int posicion)*
Descripción: método que añade una nueva BaseName en una posición concreta de la lista.
Parámetros:
 - *nombre*: nombre de la nueva BaseName.
 - *Posición*: posición de la lista en la que se desea guardar el nuevo elemento.Return: void

OccurrenceXML

Propósito

El propósito de la clase Occurrence es implementar los requisitos software: RSFU -15, RSFU -16, RSFU -17, RSFU -18.

Dependencias

Tiene dependencias internas con las clases ResourceDataXML, ResourceRefXML e InstanceOfXML. Tiene dependencias externas con el componente GestionScope y con la librería betwixt, que permite grabar y recuperar la información en formato XML.

Interfaces

La clase Occurrence dispone de las siguientes operaciones. Los get y los set de los atributos son métodos que no se describirán:

- *dameResourceRef()*
Descripción: método de acceso a los ResourceRef del Occurrence.
Parámetros: void
Return: lista con los ResourceRef del Occurrence seleccionado.
- *dameResourceData()*
Descripción: método de acceso a los ResourceData del Occurrence.
Parámetros: void
Return: lista con los ResourceData del Occurrence seleccionado.
- *addResourceRef(ResourceRefXML resourceRef)*
Descripción: método que añade un nuevo ResourceRef al Occurrence.
Parámetros:
 - *resourceRef*: nuevo ResourceRef del Occurrence.Return: void
- *getResourceRefIterator()*
Descripción: accedente al iterador de la lista de ResourceRef del Occurrence.
Parámetros: void
Return: el iterador de la lista.
- *addResourceData(String resourceData)*
Descripción: método que añade un nuevo ResourceData al Occurrence.
Parámetros:
 - *resourceData*: nuevo ResourceData del Occurrence.Return: void
- *getResourceDataIterator()*
Descripción: accedente al iterador de la lista de ResourceData del Occurrence.
Parámetros: void
Return: el iterador de la lista.



ResourceDataXML

Propósito

El propósito de la clase ResourceDataXML es implementar los requisitos software: RSFU - 15, RSFU -16, RSFU -17, RSFU -18.

Dependencias

La clase estudiada no depende de otras clases. Tiene dependencias externas con la librería betwixt, que permite grabar y recuperar la información en formato XML.

Interfaces

La clase ResourceDataXML sólo dispone de un constructor y métodos get y set de sus dos atributos id y nombre.



ResourceRefXML

Propósito

El propósito de la clase ResourceRefXML es implementar los requisitos software: RSFU -15, RSFU -16, RSFU -17, RSFU -18.

Dependencias

La clase estudiada no depende de otras clases. Tiene dependencias externas con la librería betwixt, que permite grabar y recuperar la información en formato XML.

Interfaces

La clase ResourceDataXML sólo dispone de un constructor y métodos get y set de sus atributos id, xlinType y xlinkHref.



BaseNameXML

Propósito

El propósito de la clase BaseNameXML es implementar los requisitos software: RSFU -19, RSFU -20, RSFU -21, RSFU -22.

Dependencias

La clase estudiada no tiene dependencias internas. Tiene dependencias externas con el componente GestionScope y con la librería betwixt, que permite grabar y recuperar la información en formato XML.

Interfaces

La clase BaseNameXML sólo dispone de un constructor y métodos get y set de sus atributos id, scope y baseNameString.

SubjectIdentityXML

Propósito

El propósito de la clase SubjectIdentityXML es implementar los requisitos software: RSFU - 11, RSFU -12, RSFU -13, RSFU -14.

Dependencias

La clase estudiada tiene dependencias internas con las clases SubjectIndicatorRefXML y TopicRefXML. Tiene dependencias externas con la librería betwixt, que permite grabar y recuperar la información en formato XML.

Interfaces

La clase SubjectIdentityXML dispone de las siguientes operaciones. Los get y los set de los atributos son métodos que no se describirán:

- *dameTopicRef ()*
Descripción: método de acceso a los TopicRef del SubjectIdentity.
Parámetros: void
Return: lista con los TopicRef del SubjectIdentity seleccionado.
- *dameSubjectIndicatorRef ()*
Descripción: método de acceso a los SubjectIndicatorRef del SubjectIdentity.
Parámetros: void
Return: lista con los SubjectIndicatorRef del SubjectIdentity seleccionado.
- *addTopicRef(TopicRefXML topicRef)*
Descripción: método que añade un nuevo TopicRef al SubjectIdentity.
Parámetros:
 - *topicRef*: nuevo TopicRef del SubjectIdentity.Return: void
- *getTopicRefIterator()*
Descripción: accedente al iterador de la lista de TopicRef del SubjectIdentity.
Parámetros: void
Return: el iterador de la lista.
- *addSubjectIndicatorRef(SubjectIndicatorRefXML subjectIndicatorRef)*
Descripción: método que añade un nuevo SubjectIndicatorRef al SubjectIdentity.
Parámetros:
 - *subjectIndicatorRef*: nuevo SubjectIndicatorRef del SubjectIdentity.Return: void
- *getSubjectIndicatorRefIterator()*
Descripción: accedente al iterador de la lista de SubjectIndicatorRef.
Parámetros: void
Return: el iterador de la lista.



SubjectIndicatorRefXML

Propósito

El propósito de la clase SubjectIndicatorRefXML es implementar los requisitos software: RSFU -11, RSFU -12, RSFU -13, RSFU -14.

Dependencias

La clase estudiada no tiene dependencias internas. Tiene dependencias con la librería betwixt, que permite grabar y recuperar la información en formato XML.

Interfaces

La clase SubjectIndicatorRefXML sólo dispone de un constructor y métodos get y set de sus atributos id, xlinType y xlinkHref.



TopicRefXML

Propósito

El propósito de la clase TopicRefXML es implementar los requisitos software: RSFU -11, RSFU -12, RSFU -13, RSFU -14.

Dependencias

La clase estudiada no tiene dependencias internas. Tiene dependencias con la librería betwixt, que permite grabar y recuperar la información en formato XML.

Interfaces

La clase TopicRefXML sólo dispone de un constructor y métodos get y set de sus atributos id, xlinType y xlinkHref.



InstanceOfXML

Propósito

El propósito de la clase InstanceOfXML es implementar los requisitos software: RSFU -08, RSFU -09, RSFU -10.

Dependencias

La clase estudiada tiene dependencias internas con las clases SubjectIndicatorRefXML y TopicRefXML. Tiene dependencias externas con la librería betwixt, que permite grabar y recuperar la información en formato XML.

Interfaces

La clase InstanceOfXML sólo dispone de un constructor y métodos get y set de sus atributos id, topicRef y subjectIndicatorRef.

5.2.3. Modelo.gestionAssociation

Propósito

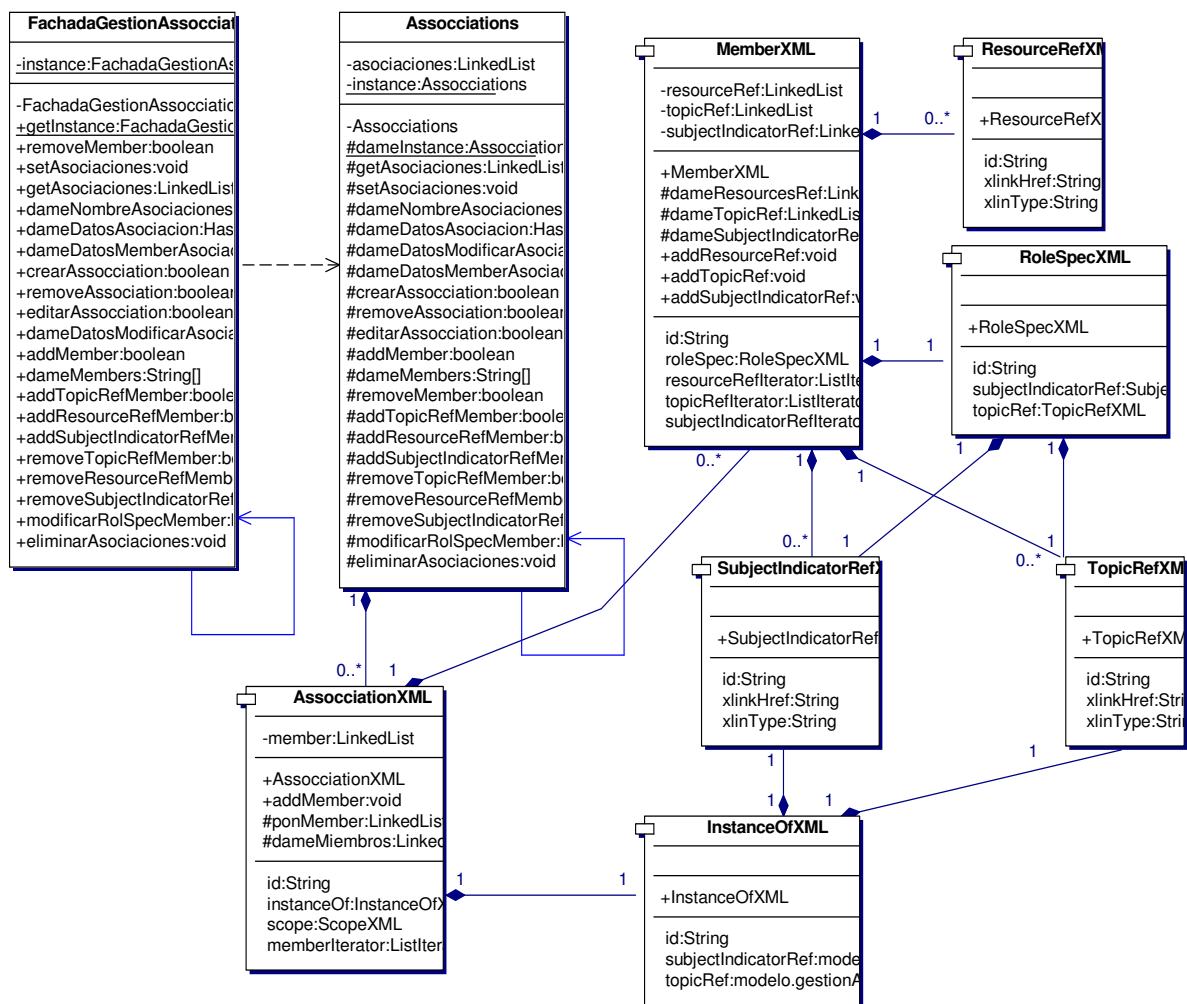
El componente gestionAssociation implementa los siguientes requisitos software: RSFU-23, RSFU -24, RSFU -25, RSFU -26, RSFU -27, RSFU -28, RSFU -29.

Dependencias

El componente gestionAssociation depende de la librería betwixt para recuperar y grabar la información en formato xtm.

Interfaces

El componente “ofrece” una interfaz de comunicación hacia el exterior (FachadaGestionAssociation), por medio de la cual los servicios prestados son accesibles para otros componentes del sistema. Los servicios están estructurados según el siguiente diagrama:



FachadaGestionAssociation

Propósito

El propósito de la clase FachadaGestionAssociation es implementar los requisitos software: RSFU-23, RSFU -24, RSFU -25, RSFU -26, RSFU -27, RSFU -28, RSFU -29. Además, la clase hará de fachada para ofrecer a los demás componentes todas las funcionalidades del componente gestionAssociation y evitar, de este modo, que las clases que formen al componente puedan ser vistas por los demás componentes.

Dependencias

La clase estudiada depende directamente de la clase Associations.

Interfaces

La FachadaGestionAssociation tiene las siguientes operaciones de “acceso” al componente:

- *removeMember(int asociacion, int indiceMember)*
Descripción: elimina el miembro de la asociación seleccionado.
Parámetros
 - *asociacion*: índice de la asociación de la que se desea borrar el miembro.
 - *indiceMember*: índice del member que se desea borrar.Return: true si todo ha salido bien, o false en caso contrario.
- *setAsociaciones(LinkedList asociaciones)*
Descripción: inserta una lista de asociaciones recuperadas del xtm.
Parámetros
 - *asociaciones*: lista de asociaciones.Return: void.
- *getAsociaciones()*
Descripción: accedente a la lista de asociaciones.
Parámetros: void.
Return: la lista de asociaciones.
- *dameNombreAsociaciones()*
Descripción: accedente al nombre de los asociaciones.
Parámetros: void.
Return: la lista con los nombres de las asociaciones.
- *dameDatosAsociacion(int asociacion)*
Descripción: recupera los datos de una asociación.
Parámetros
 - *asociacion*: índice de la asociación seleccionada.Return: hashmap con los datos de la asociación seleccionada.
- *dameDatosMemberAsociacion(int asociacion, int member)*
Descripción: recupera los datos de un miembro de la asociación seleccionada.
Parámetros



- *asociacion*: índice de la asociación seleccionada.
 - *member*: índice del miembro seleccionado.

Return: hashmap con los datos del miembro seleccionado.
- *crearAssociation(String idAsociacion, String idMember, int tipoInstance, String href, String idScope, DefaultListModel listaTopicRef, DefaultListModel listaResourceRef, DefaultListModel listaSubjectIndicatorRef)*

Descripción: crea una nueva asociación.

Parámetros

 - *idAsociacion*: id de la asociación.
 - *idMember*: id del miembro.
 - *tipoInstance*: tipo de instanteOf (TopicRef o SubjectIndicatorRef).
 - *href*: instanceOf de la asociación.
 - *idScope*: id del Scope de la Occurrence.
 - *listaTopicRef*: lista de TopicRef del Scope asociado.
 - *listaResourceRef*: lista de ResourceRef del Scope asociado.
 - *listaSubjectIndicatorRef*: lista de SubjectIndicatorRef del Scope asociado.

Return: true si se ha creado correctamente, false en caso contrario.
- *removeAssociation(int claveSelected)*

Descripción: elimina la asociación seleccionada.

Parámetros

 - *asociacion*: índice de la asociación seleccionada.

Return: true si se ha eliminado correctamente, false en caso contrario.
- *editarAssociation(int indice, String idAsociacion, int tipoInstance, String href, String idScope, DefaultListModel listaTopicRef, DefaultListModel listaResourceRef, DefaultListModel listaSubjectIndicatorRef)*

Descripción: modifica los datos de una nueva asociación.

Parámetros

 - *indice*: índice de la asociación que se desea modificar.
 - *idAsociacion*: id de la asociación.
 - *tipoInstance*: tipo de instanteOf (TopicRef o SubjectIndicatorRef).
 - *href*: instanceOf de la asociación.
 - *idScope*: id del Scope de la Occurrence.
 - *listaTopicRef*: lista de TopicRef del Scope asociado.
 - *listaResourceRef*: lista de ResourceRef del Scope asociado.
 - *listaSubjectIndicatorRef*: lista de SubjectIndicatorRef del Scope asociado.

Return: true si se ha modificado correctamente, false en caso contrario.
- *dameDatosModificarAsociacion(int claveSelected)*

Descripción: recupera los datos que se mostrarán cuando se modifica una asociación.

Parámetros

 - *claveSelected*: índice de la asociación seleccionada.

Return: hashmap con los datos de la asociación que se va a modificar.
- *addMember(String asociacion, String idMember)*

Descripción: añade un nuevo miembro a la asociación.



Parámetros

- *indice*: índice de la asociación.
- *idMember*: id del nuevo miembro.

Return: true si se ha creado correctamente, false en caso contrario.

- *dameMembers(int indice)*
Descripción: recupera los nombres de los miembros de la asociación seleccionada.
Parámetros
 - *indice*: índice de la asociación seleccionada.Return: array con los nombres de los miembros de la asociación.
- *addTopicRefMember(String asociacion, String member, String topicRef)*
Descripción: añade un nuevo TopicRef al miembro seleccionado.
Parámetros
 - *asociacion*: índice de la asociación.
 - *member*: id del nuevo miembro.
 - *topicRef*: Nuevo TopicRef del miembro.Return: true si se ha creado correctamente, false en caso contrario.
- *addResourceRefMember(String asociacion, String member, String resourceRef)*
Descripción: añade un nuevo ResourceRef al miembro seleccionado.
Parámetros
 - *asociacion*: índice de la asociación.
 - *member*: id del nuevo miembro.
 - *resourceRef*: Nuevo ResourceRef del miembro.Return: true si se ha creado correctamente, false en caso contrario.
- *addSubjectIndicatorRefMember(String asociacion, String member, String subjectIndicatorRef)*
Descripción: añade un nuevo subjectIndicatorRef al miembro seleccionado.
Parámetros
 - *asociacion*: índice de la asociación.
 - *member*: id del nuevo miembro.
 - *subjectIndicatorRef*: Nuevo subjectIndicatorRef del miembro.Return: true si se ha creado correctamente, false en caso contrario.
- *removeTopicRefMember(String asociacion, String member, String topicRef)*
Descripción: elimina un TopicRef al miembro seleccionado.
Parámetros
 - *asociacion*: índice de la asociación.
 - *member*: id del nuevo miembro.
 - *topicRef*: TopicRef que se desea eliminar.Return: true si se ha eliminado correctamente, false en caso contrario.
- *boolean removeResourceRefMember(String asociacion, String member, String resourceRef)*
Descripción: elimina un ResourceRef al miembro seleccionado.
Parámetros



- *asociacion*: índice de la asociación.
- *member*: id del nuevo miembro.
- *resourceRef*: ResourceRef que se desea eliminar.

Return: true si se ha eliminado correctamente, false en caso contrario.

- *removeSubjectIndicatorRefMember(String asociacion, String member, String subjectIndicatorRef)*

Descripción: elimina un SubjectIndicatorRef al miembro seleccionado.

Parámetros

- *asociacion*: índice de la asociación.
- *member*: id del nuevo miembro.
- *subjectIndicatorRef*: SubjectIndicatorRef que se desea eliminar.

Return: true si se ha eliminado correctamente, false en caso contrario.

- *modificarRolSpecMember(String asociacion, String member, String rolSpec, String tipoRol)*

Descripción: modifica el rol del miembro seleccionado.

Parámetros

- *asociacion*: índice de la asociación.
- *member*: id del nuevo miembro.
- *rolSpec*: nuevos datos del rol.
- *tipoRol*: tipo del rol.

Return: true si se ha modificado correctamente, false en caso contrario.

- *void eliminarAsociaciones()*

Descripción: elimina todas las asociaciones.

Parámetros: void.

Return: void.

- *getInstance();*

Descripción: método que permite al usuario acceder al componente. Gracias a este método se implementa el patrón Singleton.

Parámetros

Association

Propósito

El propósito de la clase Association es implementar los requisitos software: RSFU-23, RSFU -24, RSFU -25, RSFU -26, RSFU -27, RSFU -28, RSFU -29.

Dependencias

La clase estudiada tiene dependencias internas con la clase AssociationXML.

Interfaces

La clase Association dispone de las siguientes operaciones:

- *setAsociaciones(LinkedList asociaciones)*
Descripción: inserta una lista de asociaciones recuperadas del xtm.
Parámetros
 - *asociaciones*: lista de asociaciones.Return: void.
- *getAsociaciones()*
Descripción: accedente a la lista de asociaciones.
Parámetros: void.
Return: la lista de asociaciones.
- *dameNombreAsociaciones()*
Descripción: accedente al nombre de los asociaciones.
Parámetros: void.
Return: la lista con los nombres de las asociaciones.
- *dameDatosAsociacion(int asociacion)*
Descripción: recupera los datos de una asociación.
Parámetros
 - *asociacion*: índice de la asociación seleccionada.Return: hashmap con los datos de la asociación seleccionada.
- *dameDatosModificarAsociacion(int claveSelected)*
Descripción: recupera los datos que se mostrarán cuando se modifica una asociación.
Parámetros
 - *claveSelected*: índice de la asociación seleccionada.Return: hashmap con los datos de la asociación que se va a modificar.
- *dameDatosMemberAsociacion(int asociacion, int member)*
Descripción: recupera los datos de un miembro de la asociación seleccionada.
Parámetros
 - *asociacion*: índice de la asociación seleccionada.
 - *member*: índice del miembro seleccionado.Return: hashmap con los datos del miembro seleccionado.

- *crearAssociation(String idAsociacion, String idMember, int tipoInstance, String href, String idScope, DefaultListModel listaTopicRef, DefaultListModel listaResourceRef, DefaultListModel listaSubjectIndicatorRef)*
Descripción: crea una nueva asociación.
Parámetros
 - *idAsociacion*: id de la asociación.
 - *idMember*: id del miembro.
 - *tipoInstance*: tipo de instanteOf (TopicRef o SubjectIndicatorRef).
 - *href*: instanceOf de la asociación.
 - *idScope*: id del Scope de la Occurrence.
 - *listaTopicRef*: lista de TopicRef del Scope asociado.
 - *listaResourceRef*: lista de ResourceRef del Scope asociado.
 - *listaSubjectIndicatorRef*: lista de SubjectIndicatorRef del Scope asociado.Return: true si se ha creado correctamente, false en caso contrario.
- *removeAssociation(int claveSelected)*
Descripción: elimina la asociación seleccionada.
Parámetros
 - *asociacion*: índice de la asociación seleccionada.Return: true si se ha eliminado correctamente, false en caso contrario.
- *editarAssociation(int indice, String idAsociacion, int tipoInstance, String href, String idScope, DefaultListModel listaTopicRef, DefaultListModel listaResourceRef, DefaultListModel listaSubjectIndicatorRef)*
Descripción: modifica los datos de una nueva asociación.
Parámetros
 - *indice*: índice de la asociación que se desea modificar.
 - *idAsociacion*: id de la asociación.
 - *tipoInstance*: tipo de instanteOf (TopicRef o SubjectIndicatorRef).
 - *href*: instanceOf de la asociación.
 - *idScope*: id del Scope de la Occurrence.
 - *listaTopicRef*: lista de TopicRef del Scope asociado.
 - *listaResourceRef*: lista de ResourceRef del Scope asociado.
 - *listaSubjectIndicatorRef*: lista de SubjectIndicatorRef del Scope asociado.Return: true si se ha modificado correctamente, false en caso contrario.
- *addMember(String asociacion, String idMember)*
Descripción: añade un nuevo miembro a la asociación.
Parámetros
 - *indice*: índice de la asociación.
 - *idMember*: id del nuevo miembro.Return: true si se ha creado correctamente, false en caso contrario.
- *dameMembers(int indice)*
Descripción: recupera los nombres de los miembros de la asociación seleccionada.
Parámetros
 - *indice*: índice de la asociación seleccionada.



Return: array con los nombres de los miembros de la asociación.

- *removeMember(int asociacion, int indiceMember)*
Descripción: elimina el miembro de la asociación seleccionado.
Parámetros
 - *asociacion*: índice de la asociación de la que se desea borrar el miembro.
 - *indiceMember*: índice del member que se desea borrar.Return: true si todo ha salido bien, o false en caso contrario.
- *addTopicRefMember(String asociacion, String member, String topicRef)*
Descripción: añade un nuevo TopicRef al miembro seleccionado.
Parámetros
 - *asociacion*: índice de la asociación.
 - *member*: id del nuevo miembro.
 - *topicRef*: Nuevo TopicRef del miembro.Return: true si se ha creado correctamente, false en caso contrario.
- *addResourceRefMember(String asociacion, String member, String resourceRef)*
Descripción: añade un nuevo ResourceRef al miembro seleccionado.
Parámetros
 - *asociacion*: índice de la asociación.
 - *member*: id del nuevo miembro.
 - *resourceRef*: Nuevo ResourceRef del miembro.Return: true si se ha creado correctamente, false en caso contrario.
- *addSubjectIndicatorRefMember(String asociacion, String member, String subjectIndicatorRef)*
Descripción: añade un nuevo subjectIndicatorRef al miembro seleccionado.
Parámetros
 - *asociacion*: índice de la asociación.
 - *member*: id del nuevo miembro.
 - *subjectIndicatorRef*: Nuevo subjectIndicatorRef del miembro.Return: true si se ha creado correctamente, false en caso contrario.
- *removeTopicRefMember(String asociacion, String member, String topicRef)*
Descripción: elimina un TopicRef al miembro seleccionado.
Parámetros
 - *asociacion*: índice de la asociación.
 - *member*: id del nuevo miembro.
 - *topicRef*: TopicRef que se desea eliminar.Return: true si se ha eliminado correctamente, false en caso contrario.
- *boolean removeResourceRefMember(String asociacion, String member, String resourceRef)*
Descripción: elimina un ResourceRef al miembro seleccionado.
Parámetros
 - *asociacion*: índice de la asociación.
 - *member*: id del nuevo miembro.
 - *resourceRef*: ResourceRef que se desea eliminar.



Return: true si se ha eliminado correctamente, false en caso contrario.

- *removeSubjectIndicatorRefMember(String asociacion, String member, String subjectIndicatorRef)*
Descripción: elimina un SubjectIndicatorRef al miembro seleccionado.
Parámetros
 - *asociacion*: índice de la asociación.
 - *member*: id del nuevo miembro.
 - *subjectIndicatorRef*: SubjectIndicatorRef que se desea eliminar.Return: true si se ha eliminado correctamente, false en caso contrario.
- *modificarRolSpecMember(String asociacion, String member, String rolSpec, String tipoRol)*
Descripción: modifica el rol del miembro seleccionado.
Parámetros
 - *asociacion*: índice de la asociación.
 - *member*: id del nuevo miembro.
 - *rolSpec*: nuevos datos del rol.
 - *tipoRol*: tipo del rol.Return: true si se ha modificado correctamente, false en caso contrario.
- *void eliminarAsociaciones()*
Descripción: elimina todas las asociaciones.
Parámetros: void.
Return: void.
- *getInstance();*
Descripción: método que permite al usuario acceder al componente. Gracias a este método se implementa el patrón Singleton.
Parámetros

AssociationXML

Propósito

El propósito de la clase AssociationXML es implementar los requisitos software: RSFU-23, RSFU -24, RSFU -25, RSFU -26, RSFU -27, RSFU -28, RSFU -29.

Dependencias

La clase estudiada tiene dependencias internas con la clase InstanceOfXML y con la clase MemberXML. Además, presenta dependencias externas con la clase ScopeXML del componente GestiónScope y dependencias con la librería betwixt para guardar y recuperar los datos en formato xml.

Interfaces

La clase AssociationXML dispone de un constructor y métodos get y set de sus atributos id, instanceOf y scope, además de las operaciones descritas.

- *addMember(MemberXML member)*
Descripción: método que añade un nuevo member a la asociación.
Parámetros:
 - *member*: nuevo miembro de la asociación.Return: void
- *getMemberIterator()*
Descripción: accedente al iterador de la lista de miembro de la asociación.
Parámetros: void
Return: el iterador de la lista
- *ponMember(LinkedList miembros)*
Descripción: método que inserta una lista de miembros a la asociación.
Parámetros:
 - *miembros*: lista con los miembros de la asociación.Return: void
- *dameMiembros()*
Descripción: método de acceso a los miembros de la asociación.
Parámetros: void
Return: lista con los miembros de la asociación.



MemberXML

Propósito

El propósito de la clase MemberXML es implementar los requisitos software: RSFU -27, RSFU -28, RSFU -29.

Dependencias

La clase estudiada tiene dependencias internas con las clases RoleSpecXML, ResourceRefXML, TopicRefXML y SubjectIndicatorRefXML. Además, presenta dependencias con la librería betwixt para guardar y recuperar los datos en formato xml.

Interfaces

La clase MemberXML dispone de un constructor y métodos get y set de sus atributos id y rolSpec, además de las operaciones descritas.

- *dameResourcesRef()*
Descripción: método de acceso a los ResourceRef del member.
Parámetros: void
Return: lista con los ResourceRef del member seleccionado.
- *dameTopicRef()*
Descripción: método de acceso a los topicRef del member.
Parámetros: void
Return: lista con los topicRef del member seleccionado.
- *dameSubjectIndicatorRef()*
Descripción: método de acceso a los subjectIndicatorRef del member.
Parámetros: void
Return: lista con los subjectIndicatorRef del member seleccionado.
- *addResourceRef(ResourceRefXML resourceRef)*
Descripción: método que añade un nuevo ResourceRef al member.
Parámetros:
 - *resourceRef*: nuevo ResourceRef del member.Return: void
- *getResourceRefIterator()*
Descripción: accedente al iterador de la lista de ResourceRef del member.
Parámetros: void
Return: el iterador de la lista
- *addTopicRef(TopicRefXML topicRef)*
Descripción: método que añade un nuevo TopicRef al member.
Parámetros:
 - *topicRef*: nuevo TopicRef del member.Return: void



- *getTopicRefIterator()*
Descripción: accedente al iterador de la lista de TopicRef del member.
Parámetros: void
Return: el iterador de la lista
- *addSubjectIndicatorRef(SubjectIndicatorRefXML subjectIndicatorRef)*
Descripción: método que añade un nuevo SubjectIndicatorRef al member.
Parámetros:
 - *topicRef*: nuevo SubjectIndicatorRef del member.Return: void
- *getSubjectIndicatorRefIterator()*
Descripción: accedente al iterador de la lista de SubjectIndicatorRef del member.
Parámetros: void
Return: el iterador de la lista



RoleSpecXML

Propósito

El propósito de la clase RoleSpecXML es implementar los requisitos software: RSFU -27, RSFU -28, RSFU -29.

Dependencias

La clase estudiada tiene dependencias internas con las clases TopicRefXML y SubjectIndicatorRefXML. Además, presenta dependencias con la librería betwixt para guardar y recuperar los datos en formato xml.

Interfaces

La clase RoleSpecXML dispone de un constructor y métodos get y set de sus atributos id y topicRef y subjectIndicatorRef.



SubjectIndicatorRefXML

Propósito

El propósito de la clase SubjectIndicatorRefXML es implementar los requisitos software: RSFU -27, RSFU -28, RSFU -29.

Dependencias

La clase estudiada no tiene dependencias internas. Tiene dependencias con la librería betwixt, que permite grabar y recuperar la información en formato XML.

Interfaces

La clase SubjectIndicatorRefXML sólo dispone de un constructor y métodos get y set de sus atributos id, xlinType y xlinkHref.



TopicRefXML

Propósito

El propósito de la clase TopicRefXML es implementar los requisitos software: RSFU -27, RSFU -28, RSFU -29.

Dependencias

La clase estudiada no tiene dependencias internas. Tiene dependencias con la librería betwixt, que permite grabar y recuperar la información en formato XML.

Interfaces

La clase TopicRefXML sólo dispone de un constructor y métodos get y set de sus atributos id, xlinType y xlinkHref.



ResourceRefXML

Propósito

El propósito de la clase ResourceRefXML es implementar los requisitos software: RSFU -27, RSFU -28, RSFU -29.

Dependencias

La clase estudiada no depende de otras clases. Tiene dependencias externas con la librería betwixt, que permite grabar y recuperar la información en formato XML.

Interfaces

La clase ResourceDataXML sólo dispone de un constructor y métodos get y set de sus atributos id, xlinType y xlinkHref.

5.2.4. Modelo.gestionScope

Propósito

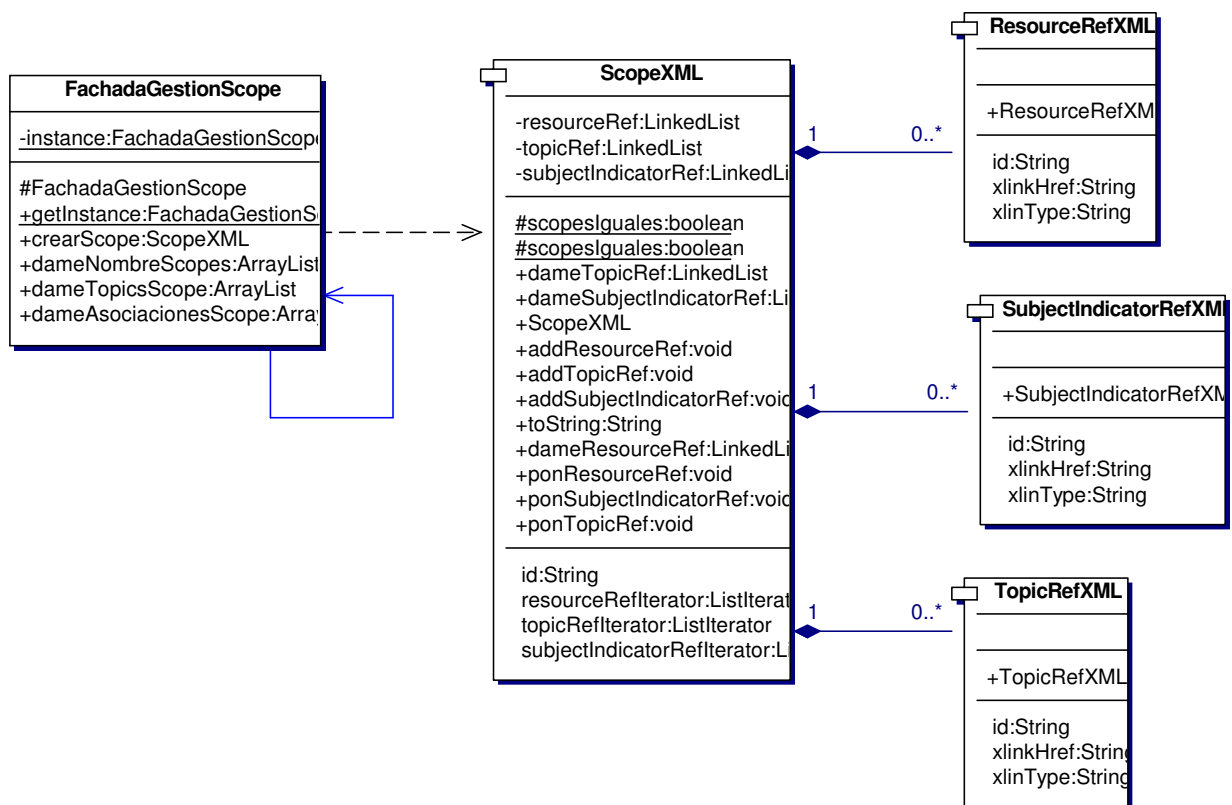
El componente gestionScope implementa los siguientes requisitos software: RSFU -18, RSFU -22, RSFU -26, RSFU -30, RSFU -31.

Dependencias

El componente gestionScope depende de la librería betwixt para recuperar y grabar la información en formato xtm.

Interfaces

El componente “ofrece” una interfaz de comunicación hacia el exterior (FachadaGestionScope), por medio de la cual los servicios prestados son accesibles para otros componentes del sistema. Los servicios están estructurados según el siguiente diagrama:





FachadaGestionScope

Propósito

El propósito de la clase FachadaGestionScope es implementar los requisitos software: RSFU -18, RSFU -22, RSFU -26, RSFU -30, RSFU -31. Además, la clase hará de fachada para ofrecer a los demás componentes todas las funcionalidades del componente gestionScope y evitar, de este modo, que las clases que formen al componente puedan ser vistas por los demás componentes.

Dependencias

La clase estudiada depende directamente de la clase ScopeXML.

Interfaces

La FachadaGestionScope tiene las siguientes operaciones de “acceso” al componente:

- *dameNombreScopes()*
Descripción: método de acceso a los distintos scopes creados en el TopicMap.
Parámetros: void
Return: lista con los nombres de los Scopes.
- *dameTopicsScope(HashMap scopeSelected)*
Descripción: método de acceso a los Topics del Scope.
Parámetros:
 - *scopeSelected*: datos del scope seleccionadoReturn: lista con los nombres de los Topics que pertenecen al Scope.
- *dameAsociacionesScope(HashMap hm)*
Descripción: método de acceso a los Topics del Scope.
Parámetros:
 - *scopeSelected*: datos del scope seleccionadoReturn: lista con los nombres de los Topics que pertenecen al Scope.
- *getInstance();*
Descripción: método que permite al usuario acceder al componente. Gracias a este método se implementa el patrón Singleton.
Parámetros: no tiene.

ScopeXML

Propósito

El propósito de la clase Scope es implementar los requisitos software: RSFU -18, RSFU -22, RSFU -26, RSFU -30, RSFU -31.

Dependencias

La clase estudiada tiene dependencias internas con las clases ResourceRefXML, SubjectIndicatorRefXML y TopicRefXML. Además, tiene dependencias externas con la librería betwixt para recuperar y grabar los datos en formato xml.

Interfaces

La clase ScopeXML dispone de las siguientes operaciones. Los get y los set de los atributos son métodos que no se describirán:

- *dameResourcesRef()*
Descripción: método de acceso a los ResourceRef del Scope.
Parámetros: void
Return: lista con los ResourceRef del Scope seleccionado.
- *dameTopicRef()*
Descripción: método de acceso a los topicRef del Scope.
Parámetros: void
Return: lista con los topicRef del Scope seleccionado.
- *dameSubjectIndicatorRef()*
Descripción: método de acceso a los subjectIndicatorRef del Scope.
Parámetros: void
Return: lista con los subjectIndicatorRef del Scope seleccionado.
- *addResourceRef(ResourceRefXML resourceRef)*
Descripción: método que añade un nuevo ResourceRef al Scope.
Parámetros:
 - *resourceRef*: nuevo ResourceRef del Scope.Return: void
- *getResourceRefIterator()*
Descripción: accedente al iterador de la lista de ResourceRef del Scope.
Parámetros: void
Return: el iterador de la lista
- *addTopicRef(TopicRefXML topicRef)*
Descripción: método que añade un nuevo TopicRef al Scope.
Parámetros:
 - *topicRef*: nuevo TopicRef del Scope.Return: void



- *getTopicRefIterator()*
Descripción: accedente al iterador de la lista de TopicRef del Scope.
Parámetros: void
Return: el iterador de la lista
- *addSubjectIndicatorRef(SubjectIndicatorRefXML subjectIndicatorRef)*
Descripción: método que añade un nuevo SubjectIndicatorRef al Scope.
Parámetros:
 - *topicRef*: nuevo SubjectIndicatorRef del Scope.Return: void
- *getSubjectIndicatorRefIterator()*
Descripción: accedente al iterador de la lista de SubjectIndicatorRef del Scope.
Parámetros: void
Return: el iterador de la lista
- *scopesIguales(ScopeXML scope1, ScopeXML scope2)*
Descripción: método que permite saber si dos scopes son iguales.
Parámetros:
 - *scope1*: primer Scope a comparar.
 - *scope2*: segundo scope a comparar.Return: true si son iguales, false en caso contrario.
- *scopesIguales(ScopeXML scope1, ArrayList scopes)*
Descripción: método que permite saber si hay algún scope igual.
Parámetros:
 - *scope1*: Scope a comparar.
 - *scopes*: lista de scopes.Return: true si hay algún scope igual, false en caso contrario.
- *void ponResourceRef(LinkedList resourceRef)*
Descripción: mutador de la lista de ResourceRef del Scope.
Parámetros:
 - *resourceRef*: nueva lista de ResourceRef.Return: void
- *void ponSubjectIndicatorRef(LinkedList subjectIndicatorRef)*
Descripción: mutador de la lista de *SubjectIndicatorRef* del Scope.
Parámetros:
 - *subjectIndicatorRef*: nueva lista de *SubjectIndicatorRef*.Return: void
- *void ponTopicRef(LinkedList topicRef)*
Descripción: mutador de la lista de TopicRef del Scope.
Parámetros:
 - *topicRef*: nueva lista de TopicRef.Return: void



SubjectIndicatorRefXML

Propósito

El propósito de la clase SubjectIndicatorRefXML es implementar los requisitos software: RSFU -18, RSFU -22, RSFU -26, RSFU -30, RSFU -31.

Dependencias

La clase estudiada no tiene dependencias internas. Tiene dependencias con la librería betwixt, que permite grabar y recuperar la información en formato XML.

Interfaces

La clase SubjectIndicatorRefXML sólo dispone de un constructor y métodos get y set de sus atributos id, xlinType y xlinkHref.



TopicRefXML

Propósito

El propósito de la clase TopicRefXML es implementar los requisitos software: RSFU -18, RSFU -22, RSFU -26, RSFU -30, RSFU -31.

Dependencias

La clase estudiada no tiene dependencias internas. Tiene dependencias con la librería betwixt, que permite grabar y recuperar la información en formato XML.

Interfaces

La clase TopicRefXML sólo dispone de un constructor y métodos get y set de sus atributos id, xlinType y xlinkHref.



ResourceRefXML

Propósito

El propósito de la clase ResourceRefXML es implementar los requisitos software RSFU -18, RSFU -22, RSFU -26, RSFU -30, RSFU -31.

Dependencias

La clase estudiada no depende de otras clases. Tiene dependencias externas con la librería betwixt, que permite grabar y recuperar la información en formato XML.

Interfaces

La clase ResourceDataXML sólo dispone de un constructor y métodos get y set de sus atributos id, xlinType y xlinkHref.

5.2.5. Persistencia

Propósito

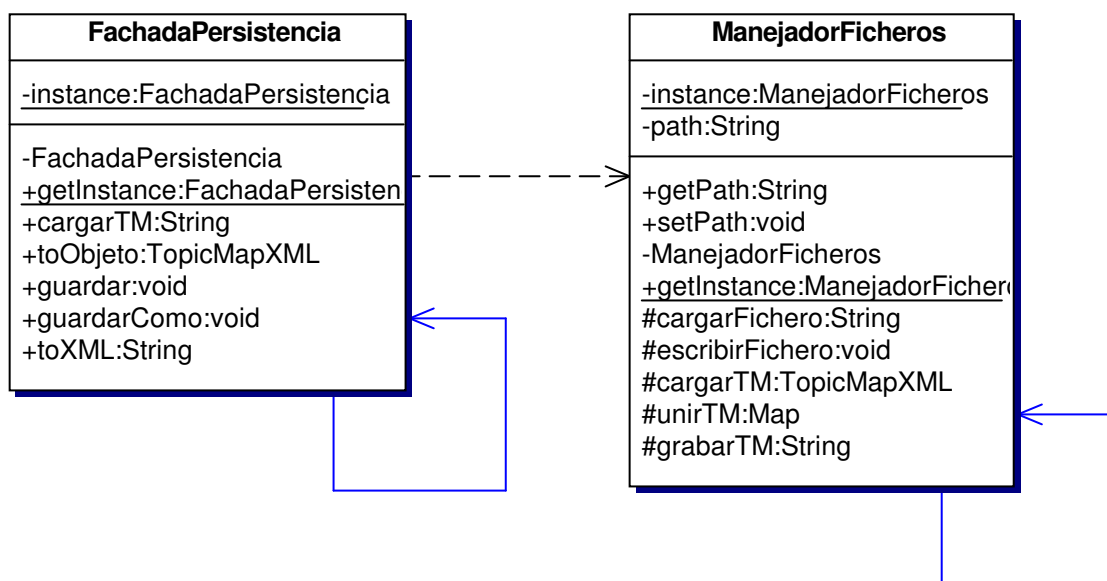
El componente persistencia implementa los siguientes requisitos software: RSFU -02, RSFU -03.

Dependencias

El componente persistencia hace uso de la librería betwixt para rescatar y cargar datos en formato xml.

Interfaces

El componente “ofrece” una interfaz de comunicación hacia el exterior (FachadaPersistencia), por medio de la cual los servicios prestados son accesibles por otros componentes del sistema. Los servicios están estructurados según el siguiente diagrama:



FachadaPersistencia

Propósito

El propósito de la clase FachadaPersistencia es implementar los requisitos software: RSFU - 02, RSFU -03. Además, la clase hará de fachada para ofrecer a los demás componentes todas las funcionalidades del componente gestionAssociation y evitar, de este modo, que las clases que formen al componente puedan ser vistas por los demás componentes.

Dependencias

La clase estudiada depende directamente de la clase ManejadorFicheros.

Interfaces

La clase FachadaPersistencia tiene las siguientes operaciones de “acceso” al componente:

- *cargarTM(String path)*
Descripción: método que permite al usuario cargar un Topic Map a partir de un fichero en formato xtm.
Parámetros
 - *path*: path del fichero xtm que queremos cargar.Return: cadena de texto con los datos cargados.
- *toObjeto(String xml)*
Descripción: método que permite crear un objeto TopicMap a partir de un xml.
Parámetros
 - *xml*: datos del TopicMap en formato xml.Return: objeto TopicMapXML con los datos cargados.
- *guardar(String xml)*
Descripción: método que permite al usuario guardar el Topic Map en formato xtm.
Parámetros
 - *xml*: xml que se desea rescatar.Return: void.
- *guardarComo(String path, String xml)*
Descripción: método que permite al usuario guardar el Topic Map en formato xtm en una url específica.
Parámetros
 - *path*: path del disco donde se desea guardar el xtm.
 - *xml*: xml que se desea rescatar.Return: void.
- *toXML(TopicMapXML topicMap)*
Descripción: método que permite parsear un objeto TopicMapXML a xml.
Parámetros
 - *topicMap*: objeto TopicMapXML.Return: datos del TopicMap en formato xtm.



- *getInstance();*
Descripción: método que permite al usuario acceder al componente. Gracias a este método se implementa el patrón Singleton.
Parámetros: no tiene.



ManejadorFicheros

Propósito

El propósito de la clase ManejadorFicheros es implementar los requisitos software: RSFU - 02, RSFU -03.

Dependencias

La clase estudiada depende de la librería betwixt para recuperar y grabar la información en formato xml.

Interfaces

La clase ManejadorFicheros está compuesta de las siguientes operaciones. Los get y los set de los atributos no serán descritos.

- *cargarFichero(String path)*
Descripción: método que permite al usuario cargar el Topic Map de un fichero xml.
Parámetros
 - *path*: path del fichero xml.Return: los datos del fichero xml.
- *escribirFichero(String path,String datosXML)*
Descripción: método que permite al usuario escribir los datos del TopicMap en un fichero xml.
Parámetros
 - *path*: path del fichero xml.
 - *datosXML*. Datos que se desean rescatar.Return: void.
- *TopicMapXML cargarTM(String xml)*
Descripción: método que permite crear un objeto TopicMap a partir de un xml.
Parámetros
 - *xml*: datos del TopicMap en formato xml.Return: objeto TopicMapXML con los datos cargados.
- *String grabarTM(TopicMapXML topicMap)*
Descripción: método que permite parsear un objeto TopicMapXML a xml.
Parámetros
 - *topicMap*: objeto TopicMapXML.Return: datos del TopicMap en formato xtm.
- *getInstance();*
Descripción: método que permite al usuario acceder al componente. Gracias a este método se implementa el patrón Singleton.
Parámetros: no tiene.

6. Planificación

El objetivo de la planificación es definir y preparar las condiciones de trabajo del equipo. Para ello, se establecen fechas previstas para la realización del conjunto de actividades que lo componen, teniendo en cuenta que se deben emplear para ello unos recursos que llevan implícitos unos costes.

Para cubrir el desarrollo completo del sistema se propone una serie de procesos. Antes de comenzar el proyecto, éste se divide en partes más pequeñas que puedan manejarse con mayor facilidad.

El comienzo es, por tanto, seleccionar la estructura del proyecto, estableciendo los procesos principales de desarrollo que lo integran. Para cada proceso se determinan las actividades y tareas a realizar.

A continuación se indican los procesos a desarrollar para el sistema:

1. Introducción
2. Estado del arte.
3. Herramientas
4. Análisis
5. Diseño arquitectónico
6. Diseño detallado
7. Planificación y presupuesto
8. Programación
9. Conclusiones

Para disponer de una visión amplia y general de la planificación, se ha utilizado Microsoft Project 2003, mediante el cual se ha podido generar el siguiente diagrama de Gantt donde se indica la distribución de las actividades en un calendario.

A continuación se expondrá la planificación inicial del proyecto:

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	Introducción	3 días	lun 11/06/07	mié 13/06/07	
2	<input checked="" type="checkbox"/> Estado del arte	14 días	lun 11/06/07	jue 28/06/07	
3	Topic Map	5 días	lun 11/06/07	vie 15/06/07	
4	Estándar de Topics Map	3 días	lun 18/06/07	mié 20/06/07	3
5	Estudio de mercado	6 días	jue 21/06/07	jue 28/06/07	4
6	<input checked="" type="checkbox"/> Herramientas	4 días	vie 29/06/07	mié 04/07/07	5
7	Software	2 días	vie 29/06/07	lun 02/07/07	5
8	Hardware	2 días	mar 03/07/07	mié 04/07/07	7
9	<input checked="" type="checkbox"/> Análisis	15 días	jue 05/07/07	mié 25/07/07	8
10	Requisitos de usuario	5 días	jue 05/07/07	mié 11/07/07	
11	Modelo de casos de uso	3 días	jue 12/07/07	lun 16/07/07	10
12	Requisitos software	5 días	mar 17/07/07	lun 23/07/07	11
13	Diagrama de clases inicial	2 días	mar 24/07/07	mié 25/07/07	12
14	<input checked="" type="checkbox"/> Diseño arquitectónico	6 días	jue 26/07/07	jue 02/08/07	13
15	Descripción de la descripción	2 días	jue 26/07/07	vie 27/07/07	
16	Descripción de componentes	4 días	lun 30/07/07	jue 02/08/07	15
17	Diseño detallado	15 días	vie 03/08/07	jue 23/08/07	16
18	Planificación y presupuesto	4 días	vie 24/08/07	mié 29/08/07	17
19	Programación	35 días	lun 27/08/07	vie 12/10/07	17
20	Conclusiones finales	3 días	lun 15/10/07	mié 17/10/07	19

Figura 10. Tareas y tiempos planificados inicialmente



Proyecto Fin de Carrera-TMCreator

A continuación se va a exponer el diagrama de Gantt del proyecto:

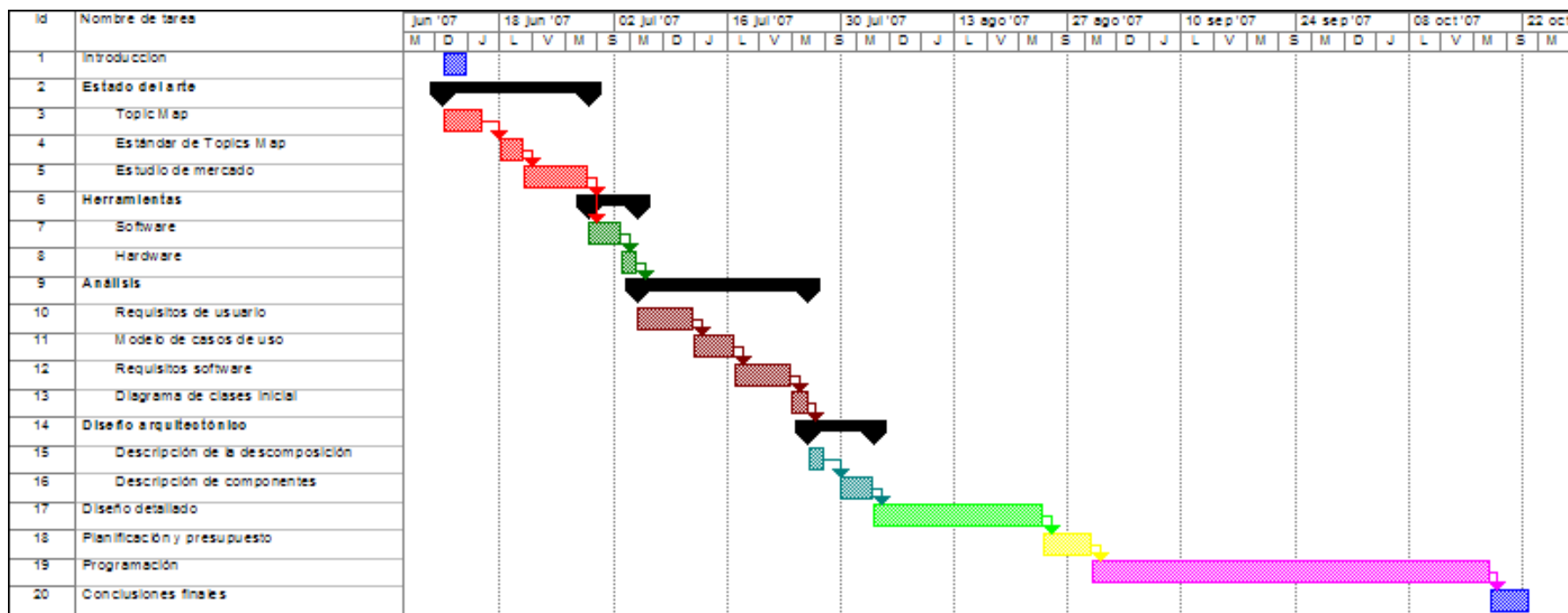


Figura 11. Diagrama de Gantt inicial

La planificación final del proyecto ha sufrido variaciones respecto a la inicial. A continuación, se expondrá una tabla resumen de las tareas realizadas y el tiempo empleado:

Diagrama de Gantt

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	Introduccion	3 días	lun 11/06/07	mié 13/06/07	
2	<input type="checkbox"/> Estado del arte	25 días	lun 11/06/07	vie 13/07/07	
3	Topic Map	10 días	lun 11/06/07	vie 22/06/07	
4	Estándar de Topics Map	5 días	lun 25/06/07	vie 29/06/07	3
5	Estudio de mercado	10 días	lun 02/07/07	vie 13/07/07	4
6	<input type="checkbox"/> Herramientas	4 días	lun 16/07/07	jue 19/07/07	5
7	Software	2 días	lun 16/07/07	mar 17/07/07	5
8	Hardware	2 días	mié 18/07/07	jue 19/07/07	7
9	<input type="checkbox"/> Análisis	21 días	vie 20/07/07	vie 17/08/07	8
10	Requisitos de usuario	7 días	vie 20/07/07	lun 30/07/07	
11	Modelo de casos de uso	5 días	mar 31/07/07	lun 06/08/07	10
12	Requisitos software	7 días	mar 07/08/07	mié 15/08/07	11
13	Diagrama de clases inicial	2 días	jue 16/08/07	vie 17/08/07	12
14	<input type="checkbox"/> Diseño arquitectónico	10 días	lun 20/08/07	vie 31/08/07	13
15	Descripción de la descomposición	5 días	lun 20/08/07	vie 24/08/07	
16	Descripción de componentes	5 días	lun 27/08/07	vie 31/08/07	15
17	Diseño detallado	20 días	lun 03/09/07	vie 28/09/07	16
18	Planificación y presupuesto	4 días	sáb 08/11/08	mié 12/11/08	
19	Programación	281 días	jue 13/11/08	jue 10/12/09	18
20	Conclusiones finales	3 días	vie 11/12/09	mar 15/12/09	19

Figura 12. Tareas y tiempos reales

Como podemos observar en la tabla resumen, hay una gran diferencia de tiempo entre la estimación inicial y la final, sobre todo desde septiembre del 2007. La planificación inicial se pensó para realizar el proyecto en 19 semanas de trabajo, trabajando los 7 días de la semana 8 horas diarias. A partir del mes de septiembre del 2007, las horas de dedicación al proyecto se limitaron drásticamente por motivos laborales y personales, pudiendo avanzar en el proyecto exclusivamente algunos sábados y domingos.

7. Estimación de costes

El objetivo de este punto es registrar los diferentes componentes del coste de mano de obra junto con el resto de costes que incorpora el desarrollo del software.

El apartado está formado por siete tablas que explican los diferentes costes que se van a tener en el desarrollo del software, y dos tablas resumen. La primera tabla resumen especifica los gastos que se prevén tener en cada actividad, y la segunda tabla resumen engloba todos los costes y se le añade además el riesgo, el beneficio y el IVA, dando como resultado el precio total del producto a desarrollar.

Los diferentes costes tratados en el documento son:

- Gastos de personal
- Gastos de equipos
- Gastos de material fungible
- Gastos de viajes
- Gastos de licencias
- Otros gastos

Cada uno de los puntos anteriores será tratado en los siguientes subapartados.

7.1. Gastos de Personal

En esta primera parte del cálculo de costes, relacionamos cada una de las actividades a desarrollar en el proyecto con el personal dedicado al mismo.

En primer lugar, debemos concretar que las actividades que se desarrollarán en este proyecto son las siguientes:

- 1) Introducción
- 2) Estado del arte
- 3) Herramientas
- 4) Análisis del Sistema
- 5) Diseño arquitectónico del Sistema
- 6) Diseño detallado del Sistema
- 7) Planificación y presupuesto
- 8) Programación

Se ha establecido que el proyecto durará 19 semanas, y se cobrará 30 € por hora trabajada. Por lo tanto, conociendo las horas planificadas de trabajo y su coste, podemos calcular los gastos de personal.

ACTIVIDADES	Horas	Precio	Total tarea
Introducción	5	30	150
Estado del arte	40	30	1.200
Herramientas	6	30	180
Análisis del Sistema	50	30	1.500
Diseño arquitectónico del Sistema	20	30	600
Diseño detallado del Sistema	60	30	1.800
Planificación y presupuesto	15	30	450
Programación	105	30	3.150
TOTAL COSTES	301	30	9.030

Tabla 4. Relación de costes por persona y actividad

7.2. Gastos de Equipos

El proyecto se ha realizado con 2 ordenadores y 1 impresora, que se debe tener en cuenta para contabilizar los gastos. Consideramos que cada ordenador tuvo un precio de compra de 800 €, y la impresora, de 60 €.

Ahora bien, no podemos contabilizar como coste todos los años el precio inicial del ordenador. Lo que se realiza normalmente en las empresas es amortizar su inmovilizado (en este caso, ordenadores e impresoras) en un determinado porcentaje anual, de tal manera, que prevén cuándo tendrán que renovar sus activos. Así, en el momento de la renovación, al haber contabilizado todos los años una amortización anual, podrán hacer frente económicamente a la adquisición de nuevos ordenadores e impresoras.

En este proyecto, tanto los ordenadores como las impresoras se amortizan linealmente durante 5 años (tiempo que estimamos para renovar el mobiliario informático). El precio inicial del ordenador dividido entre el total de semanas correspondientes a los cinco años, nos dará la amortización semanal. Mediante un simple cálculo, efectuaremos la proporción de esa amortización semanal que corresponde a los gastos de las 19 semanas de proyecto.

Equipos informáticos	Precio de compra	Amortización semanal	Amortización del proyecto (19 semanas)
Ordenadores personales (2)	1.600	6,15	116,92
Impresoras (1)	60	0,23	4,38
Routers (2)	100	0,38	7,31
TOTAL COSTES			128,62

Tabla 5. Gastos de equipos



7.3. Gastos de Material Fungible

En el siguiente cuadro, calculamos costes derivados del material fungible que he utilizado, tal como cartuchos de tinta y paquetes de folios:

Material	Precio unitario	Unidades	Coste total
Cartuchos de tinta	30	2	60
Paquetes de folios (500 uds.)	4	1	4
TOTAL COSTES			64

Tabla 6. Gastos de material fungible

7.4. Gastos de Viajes y Dietas

En cualquier proyecto, siempre hay gastos de viaje y de dietas que deben ser tenidos en cuenta.

Los gastos de viajes de este proyecto corresponden a la compra del abono mensual de la comunidad de Madrid durante cuatro meses. El precio del abono mensual es de 72 €.

En cuanto a las dietas, la cifra que aparece en esta partida en el cuadro siguiente se ha calculado teniendo en cuenta el precio de las comidas realizadas.

Gastos	Costes
Viajes	288
Dietas	200
TOTAL COSTES	488

Tabla 7. Gastos de viajes y dietas



7.5. Gastos de Licencias

A continuación, mostramos el cálculo de gastos de licencias, teniendo en cuenta su precio de compra y considerando que se han comprado para el proyecto en curso:

Licencias	Costes
Microsoft Office Professional 2007	400
Windows Vista Home Premium	160
Microsoft Project 2007	796
TOTAL COSTES	1.356

Tabla 8. Gastos de licencias



7.6. Otros gastos

En esta partida de gastos, durante los 4 meses que dura el proyecto. Asimismo, deben ser incluidos los gastos por suministros de agua, luz, teléfono; el coste de la banda ancha (adsl); y, por último, otros gastos que puedan surgir en el transcurso del proyecto y sean necesarios para el normal desarrollo del mismo.

Gastos	Costes
Suministros (agua, luz, teléfono)	100
Banda ancha (adsl)	150
Otros gastos	100
TOTAL COSTES	350

Tabla 9. Otros gastos

7.7. Resumen del presupuesto del proyecto

Por último, se presenta el resumen del presupuesto, para lo cual se han calculado todos los gastos anteriores.

Una vez que éstos se han contabilizado en una cantidad de 11.416,62 € debemos sumarle el margen de beneficio que queremos obtener (en nuestro caso, es de un 20%); un 15% de riesgo en caso de que surjan problemas en el transcurso del proyecto; y, por último, las tasas impositivas a pagar por todos esos gastos. Con todo ello, tenemos una cuantía de 17.239,09 € presupuestados para el cliente.

1. Gastos de personal	9.030,00
2. Equipos	128,62
3. Licencias	1.356,00
4. Fungible	64,00
5. Viajes y dietas	488,00
6. Otros	350,00
Total	11.416,62
Beneficio (20%)	2.283,32
Riesgo (15%)	1.712,49
I.V.A. (16%)	1.826,66
Total presupuesto	17.239,09

Tabla 10. Resumen del presupuesto del proyecto



8. Futuro

A continuación se van a enumerar una serie de mejoras del producto TMCreator que pueden ser afrontadas en trabajos futuros:

Búsquedas en los Topic Map: Los Topic Maps recogen gran cantidad de información, por lo que una mejora del proyecto podría ser la inclusión de un buscador de información que siga el estándar de ISO 18048: Topic Maps Query Language (TMQL).[10]

Visualización de TM: Se podría dar la posibilidad al usuario de crear Topic Map de una forma gráfica, o al menos, de poder visualizar el Topic Map que ha creado. La forma de visualización de Topic Maps más común es en forma de árbol, que va desplegando elementos dependiendo del elemento seleccionado por el usuario.

Compatibilidad con XTM 2.0: Actualmente el sistema es compatible con la lectura y escritura de ficheros xtm que sigan el estándar xtm 1.0. [11]

9. Conclusiones

Una vez concluidas todas las fases del proyecto, se revisan cada una de las pautas que se pretendían cumplir en los objetivos que se plantearon al comienzo del proyecto. En este apartado se exponen las conclusiones obtenidas a partir del trabajo realizado, valorando si se han cumplido o no dichos objetivos:

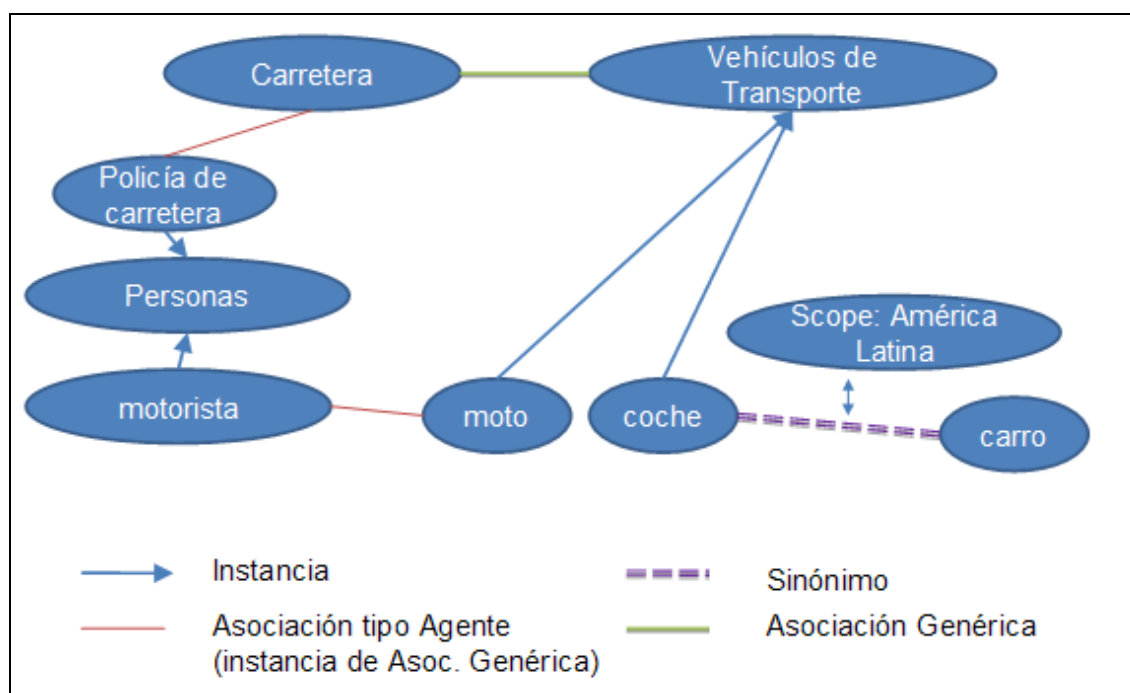
En primer lugar, se realizó un estudio exhaustivo de las aplicaciones existentes en el mercado que permitan editar y visualizar Topics Map. La mayoría de las herramientas que permiten crear Topic Map, tienen parte del trabajo automatizado ocultando información relevante al usuario que desea conocer exactamente cómo se debe crear cada elemento de Topic Map, por esta razón se estudió la posibilidad de crear la herramienta TMCreator que ayude a los usuarios a realizar Topic Map de una forma más sencilla mediante una interfaz gráfica pero que no realice ninguna operación automática que oculte información al usuario y obligue a éste a generar el fichero xtm y estudiar cómo la herramienta lo ha construido.

Finalizado el estudio de mercado, se decidió en qué tecnología se debería desarrollar el proyecto. Java fue el lenguaje elegido, principalmente por ser simple, robusto, seguro, independiente de la arquitectura, interpretado y dinámico.

Una vez terminadas las fases de análisis y diseño, se comenzó el desarrollo de la aplicación. Esta fase ha sufrido retrasos respecto a la planificación inicial. Una vez concluido el desarrollo se han realizado una serie de pruebas. Las pruebas se han realizado comparando resultados con respecto a la herramienta TM4L que es la más completa del mercado actual.

Prueba básica

La prueba básica consiste en la generación de un fichero xtm con los componentes básicos.



El resultado del fichero xtm de la prueba es el siguiente:

```
<?xml version='1.0' encoding="ISO-8859-1"?>
<topicMap xmlns="http://www.topicmaps.org/xtm/1.0/"
xmlns:xlink="http://www.w3.org/1999/xlink">
  <topic id="Vehículos">
    <baseName>
      <baseNameString>Vehículos de transporte</baseNameString>
    </baseName>
  </topic>
  <topic id="Carretera">
    <baseName>
      <baseNameString>Carretera</baseNameString>
    </baseName>
  </topic>
  <topic id="Policía">
    <instanceOf>
      <topicRef xlink:href="#Personas"/>
    </instanceOf>
    <baseName>
      <baseNameString>Policía de carretera</baseNameString>
    </baseName>
  </topic>
  <topic id="Personas">
    <baseName>
      <baseNameString>Personas</baseNameString>
    </baseName>
  </topic>
  <topic id="Motorista">
    <instanceOf>
      <topicRef xlink:href="#Personas"/>
    </instanceOf>
    <baseName>
      <baseNameString>Motorista</baseNameString>
    </baseName>
  </topic>
  <topic id="Moto">
    <instanceOf>
      <topicRef xlink:href="#Vehículos"/>
    </instanceOf>
    <baseName>
      <baseNameString>Moto</baseNameString>
    </baseName>
  </topic>
  <topic id="Coche">
    <instanceOf>
      <topicRef xlink:href="#Vehículos"/>
    </instanceOf>
    <baseName>
      <baseNameString>Coche</baseNameString>
    </baseName>
    <baseName>
      <scope id="AmericaLatina">
        <topicRef xlink:href="#ScopeAméricaLatina"/>
      </scope>
    </baseName>
  </topic>
</topicMap>
```



```
<baseNameString>Carro</baseNameString>
</baseName>
</topic>
<topic id="ScopeAméricaLatina">
  <baseName>
    <baseNameString>América Latina</baseNameString>
  </baseName>
</topic>
<topic id="TipoAsociaciónGenérica">
  <baseName>
    <baseNameString>Asociación Genérica</baseNameString>
  </baseName>
</topic>
<topic id="TipoAsociacionAgente">
  <instanceOf>
    <topicRef xlink:href="#TipoAsociaciónGenérica"/>
  </instanceOf>
  <baseName>
    <baseNameString>Asociación agente</baseNameString>
  </baseName>
</topic>
<association id="AsociaciónGenérica">
  <instanceOf>
    <topicRef xlink:href="#TipoAsociaciónGenérica"/>
  </instanceOf>
  <member id="member1">
    <topicRef xlink:href="#Vehículos"/>
  </member>
  <member id="member2">
    <topicRef xlink:href="#Carretera"/>
  </member>
</association>
<association id="AsociaciónAgente1">
  <instanceOf>
    <topicRef xlink:href="#TipoAsociacionAgente"/>
  </instanceOf>
  <member id="member3">
    <topicRef xlink:href="#Moto"/>
  </member>
  <member id="member4">
    <topicRef xlink:href="#Motorista"/>
  </member>
</association>
<association id="AsociaciónAgente2">
  <instanceOf>
    <topicRef xlink:href="#TipoAsociacionAgente"/>
  </instanceOf>
  <member id="member5">
    <topicRef xlink:href="#Carretera"/>
  </member>
  <member id="member6">
    <topicRef xlink:href="#Policía"/>
  </member>
</association>
</topicMap>
```



Prueba de compatibilidad de un fichero xtm generado con TMCreator con TM4L

El resultado generado de la prueba anterior se ha intentado abrir mediante la herramienta TM4L. La herramienta carga correctamente el fichero xtm. Se ha localizado un problema en los elementos instanceOf de los Topics, ya que estos elementos no están contemplados en TM4L; los demás componentes del Topic Map son interpretados correctamente. Al guardar los datos cargados en un nuevo fichero, las palabras con tildes se visualizan incorrectamente.

Prueba de compatibilidad de un fichero xtm generado con TMCreator con TM4L

Se ha generado un fichero con la aplicación TMCreator y se ha cargado correctamente con la herramienta TMCreator. El entendimiento de los datos cargados es bastante complicado, ya que los identificadores que TM4L genera automáticamente a los componentes de Topic Map son muy complicados.

Prueba de carga de ficheros xtm extensos

Se ha creado un fichero xtm con 150 Kb de información y se intentado cargar en la aplicación. El sistema ha sido capaz de cargar todos los datos del Topic Map en un tiempo aceptable.

Como conclusión de las pruebas anteriormente realizadas se puede afirmar que el proyecto desarrollado cumple con los requisitos inicialmente establecidos.



10. Bibliografía

[1] ISO/IEC 13250:2000. (Última visita: Julio 2007).

<http://www1.y12.doe.gov/capabilities/sgml/sc34/document/0129.pdf>

[2] ISO/IEC 13250:2003. (Última visita: Octubre 2008).

http://www1.y12.doe.gov/capabilities/sgml/sc34/document/0322_files/iso13250-2nd-ed-v2.pdf

Topic Map en la Wikipedia la enciclopedia libre.

http://en.wikipedia.org/wiki/Topic_Maps (Última visita: Septiembre 2007).

Página con información relevante de Topic Map

<http://www.topicmap.com/> (Última visita: Septiembre 2007).

[3] Página de la herramienta TMTab.

<http://www.techquila.com/tmtab/index.html> (Última visita: Octubre 2007).

[4] Página de la herramienta Topincs.

<http://www.cerny-online.com/topincs/> (Última visita: Octubre 2007).

[5] Página de la herramienta Wandora.

http://www.wandora.org/wandora/wiki/index.php?title=Main_Page (Última visita: Octubre 2007).

[6] Página de la herramienta TM4L.

<http://compsci.wssu.edu/iis/nsdl/download.html> (Última visita: Octubre 2007).

[7] Página de la herramienta TopicMap-Designer.

<http://www.topicmap-design.com/en/topicmap-designer.htm> (Última visita: Octubre 2007).

[8] Página de la herramienta Omnigator.

<http://www.ontopia.net/omnigator/models/index.jsp> (Última visita: Octubre 2007).

[9] Especificación de xtm.

<http://www.topicmaps.org/xtm/> (Última visita: Octubre 2008).

[10] Especificación TMQL.

<http://www.isotopicmaps.org/tmq/tmql.html> (Última visita: Mayo 2009).

[11] Especificación de xtm 2.0.

<http://www.isotopicmaps.org/sam/sam-xtm/> (Última visita: Mayo 2009).